2006

A fixed-point framework for launch vehicle ascent guidance

Lijun Zhang
Iowa State University

A fixed-point framework for launch vehicle ascent guidance

by

Lijun Zhang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Program of Study Committee:
Ping Lu, Major Professor
Bion L. Pierson
Thomas J. McDaniel
Ambar K. Mitra
James E. Bernard

Iowa State University

Ames, Iowa

2006

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Lijun Zhang

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Lower Case

a^*	semi-major axis of the target orbit
e^*	eccentricity of the target orbit
g_0	gravitational acceleration magnitude at sea level
$\mathbf{g}$	gravitational acceleration vector
q	dynamic pressure
$q\alpha$	product of dynamic pressure and angle of attack
$q\beta$	product of dynamic pressure and angle of sideslip
i	orbital inclination
i^*	inclination of the target orbit
$\mathbf{r}$	position vector
r	magnitude of position vector
r_f^*	launch vehicle's altitude at orbital insertion
t	time
t_f	time-to-go

t_1	time to switch from atmospheric ascent to vacuum ascent
t_{meco}	time when Main Engine Cut-off
t_{oms}	time to switch from coast to last powered stage
$\mathbf{V}$	inertial velocity vector
V_f^*	launch vehicle's inertial velocity at orbital insertion
α	vehicle angle of attack
β	vehicle angle of sideslip
η	engine throttle
η_{min}	minimum allowable engine throttle
η_0	prescribed engine throttle
γ_f^*	flight-path angle at orbital insertion
ρ	air density
θ	pitch angle
ψ	yaw angle
ϕ	roll angle
Θ	longitude for launch site
Φ	angle between $\boldsymbol{P}_v$ and $\mathbf{V}_r$
Φ_c	geocentric latitude for launch site
$\bar{\omega_e}$	Earth angular rate vector

$\mathbf{1}_b$	unit vector along the body axis X_B
$(\mathbf{1}_b)^G$	unit vector $\mathbf{1}_b$ expressed in guidance frame
$\mathbf{1}_{H0}$	unit vector toward north in ECI frame
$\mathbf{1}_{H1}$	unit vector along the direction of angular momentum in ECI frame
$\mathbf{1}_{L0}$	unit vector $\mathbf{1}_{H0}$ expressed in guidance frame
$\mathbf{1}_{L1}$	unit vector along the direction of angular momentum in guidance frame
$\mathbf{1}_n$	unit vector along normal aerodynamic force $\mathbf{N}$
$\mathbf{1}_{P_v}$	unit vector along velocity costate $\boldsymbol{P}_v$
$\mathbf{1}_{V_r}$	unit vector along relative velocity $\mathbf{V}_r$
$\mathbf{1}_y$	unit vector along the body axis Y_B
$(\mathbf{1}_y)^G$	unit vector $\mathbf{1}_y$ expressed in guidance frame
$\mathbf{1}_z$	unit vector along the body axis Z_B
$(\mathbf{1}_z)^G$	unit vector $\mathbf{1}_z$ expressed in guidance frame

Upper Case

$\mathbf{A}$	aerodynamic force in the axial direction
A_z	launch azimuth
H	Hamiltonian function
I	6 by 6 identity matrix

$I_{3\times3}$	3 by 3 identity matrix
I_{sp}	specific impulse
J	performance index
M_a	Mach number
$\mathbf{N}$	aerodynamic force in the normal direction
$\boldsymbol{P}_r$	position costate
$\boldsymbol{P}_v$	velocity costate
R_0	radius of the Earth at equator
S_ref	aerodynamic reference area
T	thrust magnitude
T_{BG}	coordinate transformation matrix from body frame to guidance frame
T_{EG}	coordinate transformation matrix from ECI to guidance frame
T_vac	vacuum thrust
V	magnitude of velocity vector
$\mathbf{V}$	velocity vector
V_r	magnitude of Earth relative velocity vector
$\mathbf{V}_r$	Earth relative velocity vector
V_s	sound velocity
Ω^*	longitude of ascending node for target orbit

Φ	angle between costate vector $\boldsymbol{P}_v$ and relative velocity $\mathbf{V}_r$
ΔT	actual thrust minus vacuum thrust

Acronyms

AMS	Analytical Multiple Shooting
ATO	Abort To Orbit
FDE	Finite Difference Equation
FD	Finite Difference
FPI	Fixed-Point Iteration
GN&C	Guidance, Navigation and Control
KSC	Kennedy Space Center, Cape Canaveral, Florida
MECO	Main Engine Cut-off
NASA	National Aeronautics and Space Administration
RLV	Reusable Launch Vehicle
RTLS	Return To Launch Site
SLI	Space Launch Initiative
SQP	Sequential Quadratic Program
SRB	Solid Rocket Booster
SSTO	Single Stage To Orbit
STS	Shuttle Transport System

TAEM	Terminal Area Energy Management
TAL	Trans-Atlantic Landing
TPBVP	Two-Point-Boundary-Value-Problem

Unit Symbols

deg	degree
kg	kilogram
km	kilometer
lbf	pound force
lbf	pound mass
m	meter
m^2	square meter
m/s	meter per second
N	Newton ($kg \cdot m/s^2$)
psf	pound per square foot (lbf/ft^2)
rad	radian
s	second
sec	second

ACKNOWLEDGEMENTS

First I would like to express my sincere thanks to my major professor Dr. Ping Lu for all of his valuable guidance, generous support and great patience over past several years.

I would also like to thank Dr. Bion L. Pierson, Dr. Thomas J. McDaniel, Dr. Ambar K. Mitra from Aerospace Engineering and Dr. James E. Bernard from Mechanical Engineering for their instruction in the class and serving as committee members on my program of study. Sincere thanks also go to Dr. Tom I-P. Shih, chair of the Department of Aerospace Engineering, for his generous encouragement on my Teaching Assistant job.

Many thanks to my former research teammate Dr. Hongsheng Sun and Dr. Hung-i Bruce Tsai for valuable discussions on the academic topics. Special acknowledgement should go to the Department of Aerospace Engineering for the wonderful and valuable education experience and the financial support for over past several years.

Finally, I would like to thank my families, especially my parents for all their unwavering support in my life. It is impossible to complete my Doctorate studies without their love and encouragement.

ABSTRACT

Recent interests in responsive launch have highlighted the need for rapid and fully automated ascent guidance planning and guidance parameter generation for launch vehicles. This dissertation aims at developing the methodology and algorithms for on-demand generation of the optimal launch vehicle ascent trajectories from lift-off to achieving targeting conditions outside the atmosphere. The entire ascent trajectory from lift-off to the final target point is partitioned into two portions: the atmospheric ascent portion and the vacuum ascent portion. The two portions are seamlessly combined together via a fixed-point iteration based on the continuity condition at the junction point between the atmospheric ascent portion and the vacuum ascent portion.

The previous research works on closed-loop endo-atmospheric ascent guidance shows that the classical finite difference method is well suited for fast solution of the constrained optimal three-dimensional ascent problem. The exploitation of certain unique features in the integration procedure between the atmospheric portion and vacuum portion and the finite difference method, allows us to cast the atmospheric ascent problem into a nested fixed-point iteration problem. Therefore a novel Fixed-Point Iteration algorithm is presented for solving the endo-atmospheric ascent guidance problem. Several approaches are also provided for facilitating the convergence of the fixed-point iteration. The exo-atmospheric ascent portion allows an optimal coast in between the two vacuum powered stages. The optimal coast enables more efficient usage of the propellant. The Analytical Multiple-Shooting algorithm is developed to find the optimal trajectory for this portion.

A generic launch vehicle model is adopted in the numerical simulation. A series of open-loop and closed-loop simulations are performed. The results verify the effectiveness, robustness and reliability of the Fixed-Point Iteration (FPI) algorithm and Analytical Multiple-Shooting (AMS) algorithm developed in this research. In comparison to Finite Difference (FD) algorithm, the Fixed-Point Iteration algorithm is more adaptive to the "cold start" case for endo-atmospheric ascent guidance. The simulations also validate the feasibility of the methodology presented in this research in rapid panning and guidance for ascent through atmosphere.

CHAPTER 1. INTRODUCTION

1.1 Background

Affordability, reliability and safety have been the key concerns for the manned space program since the first human spaceflight of Vostok 1 on April 12, 1961. As an effort to lower the cost of space flight, developing the partially reusable or even completely reusable launch vehicle has ever been a strategic goal of NASA since late 1960s. As the first orbital spacecraft designed for partial reusability, NASA's Space Shuttle, officially called Space Transportation System (STS), started in the late 1970s and has dominated NASA's manned operations since the mid-1980s. The original mission of the Space Shuttle was to operate at a high flight rate, at low cost, and with high reliability. It was intended to improve greatly on the previous generation of single-use manned and unmanned vehicles. Although it did operate as the world's first partially reusable crew-carrying spacecraft, it did not improve on those parameters in any meaningful way, and is even considered to have failed in its original purpose. It requires several thousand support personal and takes two months or more to prepare for launch. Shuttle launch costs are roughly $2,062 per pound payload, which is actually higher than most expendable launchers. About the safety concern, two Shuttles have been destroyed in 114 missions, both with the loss of the entire crew of seven: Challenger lost 73 seconds after liftoff January 28, 1986 and Columbia lost during reentry, February 1, 2003. This gives a 2 over 100 death rate per astronaut per flight. In the year of 2000, NASA unveiled its Space Launch Initiative [1], later changed to Second Generation Reusable

Launch Vehicle (RLV) program. NASA's strategic goals for a next generation RLV are to reduce the risk of crew loss to approximately 1 in 10,000 missions while lowering the cost of delivering payloads to low-Earth orbit to less than $1000 per pound. The long-term vision is to have a commercially competitive vehicle operational around the beginning of the next decade. Several specific technology objectives expected to be achieved for the RLV space vehicle, one of them is: demonstrate guidance, navigation and control systems, including autonomous flight control of checkout, takeoff, ascent, flight, reentry, and landing for an autonomously controlled space vehicle

Traditionally guidance system generates steering commands which will direct the spacecraft over vast distances from the point of departure to the destination. For a launch vehicle, guidance is part of onboard flight software that is responsible for controlling the vehicles motion so that the mission objectives, such as insertion into a specified orbit, are met while satisfying other safety and operational constraints, such as maximum bending moment or maximum allowed axial acceleration. The motion is determined by the forces acting on the vehicle which in turn are primarily a function of the vehicle attitude and throttle setting. Hence, the ascent guidance system computes the desired attitude and throttle values required to complete the mission.

The subject Ascent Guidance System determines the attitude commands and engine throttle value of the launch vehicle during the ascent, namely, the period between lift-off to final insertion into the target orbit. Whether or not the ascent trajectory is optimal can have significant impact on propellant usage for a given payload or on payload weight for the same gross vehicle weight. Therefore, the ascent guidance commands are usually optimized in some fashions. In fact, ascent guidance is one of the most notable engineering fields where optimal control theory has found routine applications.

1.2 Traditional Ascent Guidance

In ascent guidance, the complete ascent trajectory is partitioned into two different portions, atmospheric portion and vacuum portion. In the atmospheric portion, guidance system is operated in open-loop mode. Later it switches to a closed-loop vacuum guidance scheme at predetermined time or event. Here open-loop mode means that all attitude commands and engine throttle values along the flight history are predetermined. Closed-loop mode means that all attitude commands and engine throttle values are generated onboard in every guidance update cycle based on current conditions. The presence of the aerodynamic forces makes the vehicle dynamics model significantly complicated than in vacuum case where only gravity and thrust are the forces acting on the vehicle. Therefore the optimization process will be much more sensitive and computationally intensive in comparison to vacuum case. So during the atmospheric portion the optimal trajectory is calculated in pre-mission planning. In this way, all guidance commands during atmospheric flight portion are generated off-line, then updated with the day-of-launch wind data prior to launch, and loaded into the launch vehicle guidance system for use during the ascent through the atmosphere. When the vehicle reaches an altitude (for example, 50 kilometers or above) where the atmospheric density is sufficiently low, then the aerodynamic forces and wind could be neglected from the vehicle dynamics model. In this stage, a semi-analytical solution is available to determine the state and co-state variables along the optimal trajectory [7]. It makes the searching process for optimal solution very reliable and fast. Herein during vacuum ascent portion, the guidance commands could be generated through closed-loop mode. At each instant for calculating, the current state information is used as the initial condition in searching new optimal trajectory. It is why we call this guidance strategy closed-loop.

From above description, we can summarize main disadvantages of traditional ascent guidance methodology as follows. First, any mission or system parameter changes cause

re-planning of ascent guidance and it is labor-intensive and time-consuming. since all guidance commends during atmospheric ascent portion are generated off-line and loaded into the guidance system prior to launch, traditional ascent guidance lacks the adaptability to handle contingencies and abort situation in atmospheric potion. It is also not robust to deal with the situation of significant off-nominal conditions and system modeling uncertainty.

Recently, the concept of Responsive Launch had attracted attention from aerospace scientist and engineer. This concept comes from DARPA FALCON program of US Air Force [5], which is intended to realize a quick (launch within 8 hours of arrival of the payload at the launch site), and low cost access to the space. To meet such kind of requirements, the guidance system needs to determine all necessary ascent guidance parameters on a moment's notice of launch request without the need to change software. Such kind of interests in responsive launch have highlighted the need for rapid and fully automated ascent guidance planning and guidance parameter generation for launch vehicle. With continuously increment of the computer power, it will be feasible to develop the new sophisticated methodology and robust onboard algorithms for on-demand generation of optimal launch vehicle ascent trajectories.

1.3 Previous Research on Ascent Guidance

Over past several decades, different ascent guidance methods have been studied and developed. During early ages, most of them treat only the exo-atmospheric phase of flight. Brown, et. al. [11](1967), represents one of earliest attempt toward closed-loop exo-atmospheric ascent guidance. The optimal-trajectory problem is solved as a boundary value problem in ordinary differential equations to find the optimal steering laws for orbital injection and rendezvous missions. At each guidance cycle the solution to the boundary value problem is updated by a single iteration of Newton method.

Later, methods that also treat the endo-atmospheric phase have been developed. Brown, et. al. [13](1970), a linearized aerodynamic model is used to obtain the optimal thrust direction from the optimality condition. A modified shooting method is employed to solve the two-point-boundary-value problem that results from application of the necessary conditions of optimality. In order to reduce the sensitivity with respect to the initial guesses, a homotopy method is used to gradually introduce the atmospheric effects. However the reliable convergence is still not always obtained. Bradt [15](1987) use Hermite interpolation and collocation to implicitly integrate the equations of motion, and with a penalty function to reduce bending moment loads. The research aims at developing an optimal, adaptive algorithm which could adjust to changing flight conditions and performing on-board mission targeting. Cramer, et. al [16](1990) develop a nonlinear programming approach to guidance and take advantage of measured day-of-launch winds in the guidance to to provide load relief. Kelly [17](1992) develops a version of the Minimum Hamiltonian guidance algorithm which could be applied to the endo-atmospheric closed-form guidance. A generalized formulation of this algorithm was programmed with aerodynamic terms in its linearized state equations. However the similar convergence difficulties are encountered as before with Brown's method. Hanson, et. al[14](1994) investigate and test several atmospheric open-loop and closed-loop ascent guidance options for a variety of launch vehicles models. It shows a closed-loop optimal ascent guidance scheme for flying through the atmosphere is promising.

In a series of more recent work, Calise and Melamed use a hybrid collocation approach and obtain reliable convergence in dispersed guided trajectory simulations [19](1998). In References [18](1994),[20](2000), [21](2001), a hybrid approach is developed for the optimal control problem. In this approach, the analytical solution of the optimal vacuum flight and numerical collocation for atmospheric portion are combined. The solution process starts from a vacuum solution, and gradually introduces atmospheric effects and path constraint related terms until a converged solution is obtained. Possibility

of singular arcs, both for nominal ascent and aborts, is investigated too. Dukeman [22, 23](2002,2003) develops a closed-loop ascent guidance algorithm which cyclically solves the calculus-of-variations two-point boundary-value problem starting at vertical rise completion through main engine cutoff, taking into account atmospheric effects. Multiple shooting is shown to be a very effective numerical technique for this application. Lu, et. al. [7] (2003) presents a comprehensive treatment to the optimal atmospheric ascent problem subject to path constraints and final condition constraints. It is demonstrated that the classical finite difference method for two-point boundary value problems is well suited for solving the optimal ascent problem on-board.

Theoretically executing a multi-burn sequence is more efficient in fuel usage than using a single burn direct insertion. Many orbit transfer problems require the use of several burn arcs separated by relatively long optimal coast arc. In some abort cases, for example Trans-Atlantic Landing (TAL) abort, it is necessary to include a very long coast. So since the end of 1960s, a series of work has been done on multi-burn vacuum guidance problem. Brown, et al. [13](1969), Cohen and Brown [26](1972) conduct some of the earliest research on multi-burn trajectory optimization problems applied to real-time guidance problem. Jezewski [27](1972) extends previous work with linear gravity approximation to improve the computation efficiency. Gath and Calise [20, 21](2001) develop multi-burn optimization capability into a hybrid ascent guidance algorithm. Dukeman and Calise [23](2003) provide a new forms of the switching conditions which are applicable to more general final conditions.

1.4 Dissertation Overview

Recent interests in responsive launch have highlighted the need for rapid and fully automated ascent guidance planning and guidance parameter generation for launch vehicles. This dissertation aims at developing methodology and algorithms for on-demand

generation of optimal launch vehicle ascent trajectories from lift-off to achieving targeting condition outside the atmosphere. For algorithmic design the complete ascent trajectory will be divided into the atmospheric and vacuum portion. Through a novel Fixed-Point approach the endo-atmospheric ascent portion and the exo-atmospheric ascent portion are seamlessly integrated. This strategy introduces a special form of Two-Point-Boundary-Value-Problem (TPBVP) for endo-atmospheric ascent portion. An innovative Fixed-Point Algorithm is developed for solving this special TPBVP. The exo-atmospheric portion of the ascent allows an optimal coast between the two last burns. Analytical Multiple Shooting Algorithm is developed for finding the optimal trajectory for this portion. It should be emphasized that the methodology presented in this dissertation is readily applicable to the cases with more powered stages and multiple coasts in exo-atmospheric portion, provided that the number of coast arcs is specified in advance.

This dissertation is organized in 8 chapters and 2 supporting appendices.

- **Chapter 1 Introduction**: The introduction presents the background and motivation leading to the objectives and scope for this research.

- **Chapter 2 Ascent Guidance through Atmosphere**: A brief introduction to the problem of launch vehicle ascent guidance in the dissertation.

- **Chapter 3 Endo-atmospheric Ascent Guidance Formulation**: This chapter describes the dynamics modeling of the launch vehicle during atmospheric flight. The different reference frames used for dynamics modeling are introduced. Optimal control problem related to atmospheric ascent guidance is discussed. Finally the path constraints issue is addressed.

- **Chapter 4 Exo-atmospheric Ascent Guidance Formulation**: This chapter is organized in similar way with Chapter 2. The dynamics modeling and optimal control problem are presented.

- **Chapter 5 Numerical Algorithms**: This chapter describes the numerical algorithms developed in this study. A brief review of the Finite Difference (FD) algorithm is presented before introducing the Fixed-Point Iteration (FPI) algorithm.

- **Chapter 6 Verification and Validation**: Test results are summarized to verify and validate the proposed methodology and algorithms.

- **Chapter 7 Further Comparison between FPI and FD Algorithms**: Advantages and disadvantages of FPI algorithm in comparison to FD algorithm are summarized.

- **Chapter 8 Conclusions and Future Work**: This chapter gives conclusions and suggestions for further research.

1.5 Main Contributions

Following is a brief review of main contributions of this dissertation.

- A Fixed-Point framework for launch vehicle ascent guidance has been developed, tested and verified in this research. Through partitioning the complete ascent trajectory into endo-atmospheric and vacuum portions, a special two-point boundary-value problem is formulated for endo-atmospheric portion. Such kind of setting also enables to utilize the well known semi-analytical state solution and analytical costate solution in vacuum ascent guidance. The optimal trajectory for these two portions are obtained separately with different ascent guidance algorithms and seamlessly combined together through a fixed-point iteration on the state variables at the junction point between the two portions.

- A novel Fixed-Point Iteration algorithm is presented to solve the special TPBVP during atmospheric ascent portion. Special concern is given to the convergence

issue of the fixed-point iteration. Several different approaches are developed to enable and accelerate the convergence of the fixed-point iterations.

Simplicity, reliability and robustness are of foremost importance for the on-board guidance algorithm. The presented Fixed-Point Iteration algorithm employs a series fixed-point iteration to solve the special two-point boundary-value problem formulated for the endo-atmospheric ascent portion. The logic and structure of this algorithm is very simple and it is very convenient for the software implementation. The numerical simulations also demonstrate the reliability and convergence quality of this algorithm. In comparison to the existing Finite Difference (FD) method, the Fixed-Point Iteration approach is more robust in "cold start" situation. In solving the special two-point boundary-value problem of the endo-atmospheric ascent portion, the Fixed-Point Iteration algorithm presents the robustness advantage on the computation convergence when the initial guess is far away from the final converged solution. Therefore it could be expected that the presented Fixed-Point Iteration algorithm would be more reliable and robust than the existing Finite Difference algorithm in a realistic setting where it would be very difficult, if not possible, to directly obtain an initial guess that is closed to the final converged solution.

- Analytical Multiple Shooting method is developed to solve the multi-burn vacuum guidance problem with an optimal coast. The methodology presented in this research is readily applicable to the cases with more powered stages and multiple coasts.

  Since the introduction of the optimal coast, the launch vehicle is more efficient on fuel usage. Therefore for the same gross mass cases, the launch vehicle can send a larger payload into the desired orbit in the case of an optimal coast added into the ascent portion. The simulations show that the difference on payload

can be up to around 1200 kilogram for some cases with and without optimal coast. In previous related work, the multiple-shooting strategy is also adopted for mitigating the initial guess sensitivity. However they use the numerical approach to find the solution of state and costate along the optimal trajectory. In this study the linear gravity approximation is introduced and therefore enable to utilize the existing semi-analytical solution of state and the analytical solution of costate for the vacuum guidance in dealing with multi-burn trajectory optimization, which in turn improves the reliability and the computation efficiency in optimal trajectory calculation for the exo-atmospheric ascent portion.

- Capability and feasibility of the developed algorithm for on-board closed-loop ascent guidance is illustrated by a series of carefully designed closed-loop simulations.

CHAPTER 2. ASCENT GUIDANCE THROUGH ATMOSPHERE

This chapter gives a brief introduction to the problem of ascent guidance through atmosphere, which is the main topic of the dissertation.

As mentioned in Chapter 1, the complete ascent trajectory of a launch vehicle is partitioned into two portions: the atmospheric portion and the vacuum portion. The launch vehicle's ascent guidance system is performed in open-loop mode inside the dense atmosphere, and then in closed-loop mode once the vehicle clears the atmosphere. The open-loop guidance approach inherently lacks the adaptive capability to handle the contingencies and aborts, even with extensive offline planning at great costs. It also does not possess the robustness necessary to cope with significant off-nominal conditions and system modeling uncertainty, especially for new launch vehicles for which little or no flight data are available. The required re-planning and regeneration of the open-loop ascent guidance commands whenever any mission or system parameters change are costly in both developmental and operational phases of the launch vehicle. A closed-loop ascent guidance algorithm could address all of the stated deficiencies of open-loop guidance. [7]

In this dissertation, the complete ascent trajectory of a launch vehicle is still separated into two portions: endo-atmospheric (atmospheric) portion and exo-atmospheric (vacuum) portion (See Figure 2.1). The two portions are integrated together seamlessly through a fixed-point approach. In comparison to the traditional ascent guidance, an

important improvement is that the ascent guidance is performed in the closed-loop mode in both of the endo-atmospheric ascent and exo-atmospheric.

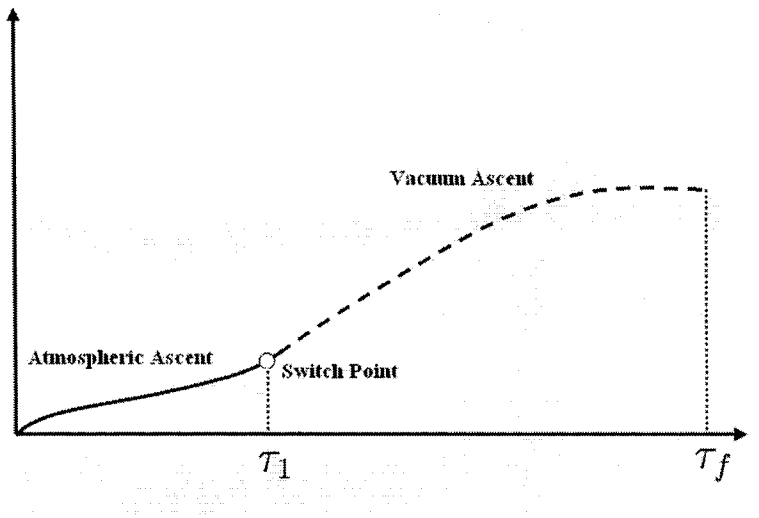

Figure 2.1 Complete ascent trajectory of a launch vehicle

2.1 Endo-atmospheric Ascent

The endo-atmospheric ascent trajectory is from lift-off at launch pad to a sufficiently high altitude where the dynamic pressure is ignorable for guidance purpose. For a generic launch vehicle, it is the first powered stage. The burn time of this portion is fixed by the propellant carried and mass flow rate of the engine. In the case of the Space Shuttle, for instance, the first powered stage will be the entire stack until the separation of the Solid Booster Rockets at 126 seconds after launch.

During endo-atmospheric portion, the presence of the aerodynamic forces in vehicle's dynamics modeling significantly complicates the solution process required for the optimal control problem, posing great difficulty to reliability and speed of convergence of the numerical process. The detailed problem formulation for the endo-atmospheric ascent is

given in Chapter 3 of the dissertation. The specially designed numerical algorithms for the endo-atmospheric ascent and the related convergence issue are carefully discussed in Chapter 5.

2.2 Exo-atmospheric Ascent

Because of the absence of aerodynamic forces in vacuum flight, the exo-atmospheric guidance problem is simpler than the case of endo-atmospheric ascent with the aid of analytical results and approximations not available to atmospheric trajectories. The complete problem formulation for the exo-atmospheric ascent portion can be found in Chapter 4.

Theoretically executing the multiple-burn sequence is more efficient on fuel usage. Therefore an optimal coast is included in between the two vacuum burns of the exo-atmospheric ascent. In the case of the Space Shuttle, it clears the atmosphere with the end of the "first powered stage". The "first powered stage" is followed by the "second powered stage" until Main Engine Cut-Off (MECO). After that, there is a coast. The coast is followed by the "third (or last) powered stage" where the Space Shuttle is powered by the Orbital Maneuver System (OMS). "analytical multiple-shooting" algorithm is presented in which the numerical setup is within the framework of conventional multiple-shooting formulation, but the propagation of the state/costate is all analytical. Detailed information is available in Chapter 5. This setting is chosen to address the sensitivity problem when the vacuum algorithm is combined with the atmospheric algorithm for the complete ascent trajectory.

2.3 Integration of Endo-atmospheric and Exo-atmospheric Ascent

One challenge in the approach of combining the atmospheric ascent algorithm with the vacuum ascent algorithm lies in the integration, when the vacuum algorithm is not just a special case of the atmospheric algorithm with zero atmospheric density. Instead of solving the entire problem with one algorithm, each of the algorithms solves one part of an optimal control problem and the trajectories in each part are then pieced together. This division, however, poses a unique problem for the atmospheric ascent algorithm because the end conditions for the atmospheric portion of the ascent are not defined a priori. Rather, they are a result of the final solution. For instance, the usual multiple-shooting formulation at the junction point will have difficulty since two different algorithms are at work, each covering only one part of the entire trajectory. Another challenge arises from the well-known fact that allowing appropriate coast between the upper stages can improve propellant consumption (or payload mass) significantly. But the presence of coast arcs, typically outside the dense atmosphere if they do occur, renders the optimal control problem more sensitive and increases the difficulty in achieving convergence, especially when the convergence of the vacuum algorithm is tied to that of the atmospheric algorithm as outlined above.

CHAPTER 3. ENDO-ATMOSPHERIC ASCENT GUIDANCE FORMULATION

In this chapter, the problem formulation of the endo-atmospheric ascent guidance will be given in details.

First, the dynamics modeling and the coordinator systems required for the three dimensional atmospheric ascent guidance are presented. The equations of motion is presented for the atmospheric flight of a launch vehicle. Three types of different coordinator systems are introduced for dynamics modeling of the launch vehicle in the atmospheric flight. The transformations between the different reference frames are provided. The way to find the Eular angles (Yaw, Pitch and Roll angles) is presented too.

After that, the optimal control problem related to the atmospheric ascent guidance is discussed. More explicitly, the issues about performance index, definition of Hamiltonian function and the derived necessary conditions for trajectory optimization are developed in details.

Path constraints are the critical concern for vehicle's atmospheric flight. The investigation on this issue will be shown in the final section of this chapter.

3.1 Atmospheric Equations of Motion of Launch Vehicle

In a Newtonian central gravitational field, referred to an inertial coordinate system, the equations of motion for a rocket-powered launch vehicle in atmospheric flight could be written as:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{V} \\ \dot{\mathbf{V}} = \mathbf{g}(\mathbf{r}) + \dfrac{T\mathbf{1}_b}{m(t)} + \dfrac{\mathbf{A}}{m(t)} + \dfrac{\mathbf{N}}{m(t)} \\ \dot{m} = -\dfrac{\eta T_{vac}}{g_0 I_{sp}} \end{cases} \tag{3.1}$$

where $\mathbf{r}$ is the vehicle's position vector from the center of the Earth to the vehicle center of gravity. $\mathbf{V}$ is the vehicle's inertial velocity vectors; $\mathbf{g}$ is the gravitational acceleration vector which is a function of $\mathbf{r}$; g_0 represents the gravitational acceleration magnitude at the radius r_0, the radius of the Earth at equator. T is the current thrust magnitude taking into account the throttle modulation and any back pressure terms; T_{vac} is the full vacuum thrust magnitude. $\mathbf{A}$ and $\mathbf{N}$ are the axial and normal aerodynamic forces respectively; $\mathbf{1}_b$ is the unit vector aligning with the RLV body longitudinal axis. $m(t)$ is the mass of the launch vehicle at current time t; η is the engine throttle setting; and I_{sp} is the specific impulse of the vehicle's engine.

3.2 Nondimensionalized Equations of Motion

In order to get a better numerical conditioning during computation process, the nondimensionalization is applied in following way:

- The distances are normalized by r_0
- Time is normalize by $\sqrt{r_0/g_0}$
- The velocity is normalized by $\sqrt{r_0 g_0}$ (which is the circular velocity around the Earth at r_0)

The vehicle's mass m could be treated as a prescribed function of time instead of a state, when the throttle setting is not a control variable to be determined. Therefore

the dimensionless equations of motion will be:

$$\begin{cases} \mathbf{r}' = \mathbf{V} \\ \mathbf{V}' = -\dfrac{1}{r^3}\mathbf{r} + (T - A)\mathbf{1}_b + N\mathbf{1}_n \end{cases} \tag{3.2}$$

where the differentiation is with respect to the dimensionless time. A and N are magnitudes of the axial and normal aerodynamic accelerations in g_0 respectively. T is the magnitude of the thrust acceleration in g_0. A, N and T are given by:

$$A = \frac{r_0}{2m(t)}\rho(r)V_r^2 S_{ref} C_A(M_a, \alpha) \tag{3.3}$$

$$N = \frac{r_0}{2m(t)}\rho(r)V_r^2 S_{ref} C_N(M_a, \alpha) \tag{3.4}$$

$$T = [\eta T_{vac} + \Delta T(r)]/m(t)g_0 \tag{3.5}$$

where, $\rho(r)$ is the dimensional atmospheric density at radius r; V_r is magnitude of the vehicle's air-relative velocity which is given by

$$\mathbf{V}_r = \mathbf{V} - \bar{\omega}_E \times \mathbf{r} - \mathbf{V}_w \tag{3.6}$$

where $\mathbf{V}_w$ is the velocity of the air relative to a frame fixed in the rotating Earth. $\bar{\omega_E}$ is the Earth angular rotation rate vector. The axial and normal aerodynamic coefficients C_A and C_N are functions of Mach number (M_a) and angle of attack (α). They are expressed in analytical forms by least squares curve fits of US 1976 standard atmosphere data.

$$C_A = C_A(\alpha, M_a) = C_{A0}(M_a) + C_{A1}(M_a)\alpha + C_{A2}(M_a)\alpha^2 \tag{3.7}$$

$$C_N = C_A(\alpha, M_a) = C_{N0}(M_a) + C_{N1}(M_a)\alpha + C_{N2}(M_a)\alpha^2 \tag{3.8}$$

The back pressure terms, represented by ΔT, are calculated with cubic spline fit. The cubic spline allows for modeling of both nonlinear back pressure terms associated with linear aerospike engines and simpler nearly exponential back pressure terms associated with conventional bell nozzle engines [42]. In all cases, curve fit parameters and cubic splines coefficients are calculated off-line.

3.3 Reference Frames Used in Dynamics Modeling

There are three kind of reference frames used in this dissertation: the Earth Centered Inertial frame (ECI), the guidance frame, and the vehicle body frame. A brief introduction of these three coordinate systems are presented in this section. The transform equations between two different coordinate systems are provided. The determination of the Eular angles is shown at the end of the section.

3.3.1 Earth Centered Inertial Reference Frame $X_I Y_I Z_I$

Earth Centered Inertial (ECI) coordinator system has its origin at the center of the Earth. The X_I points to the intersection of the Equator and the Greenwich Meridian. Z_I points to the North Pole. The direction of Y_I is determined via right-hand rule. The ECI reference frame is shown in Figure 3.1.

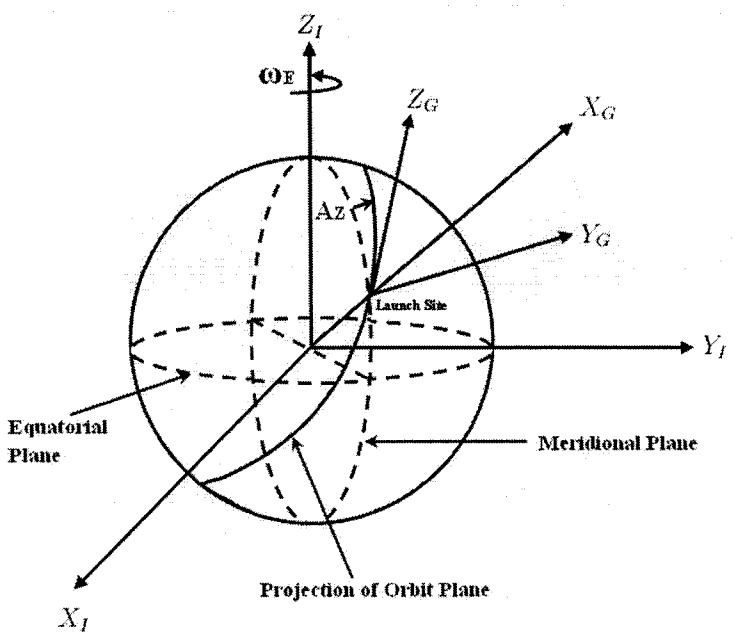

Figure 3.1 Earth centered inertial and guidance coordinator systems

3.3.2 Guidance Frame $X_G Y_G Z_G$

The guidance frame is shown in Figure 3.1 where the origin is the center of the Earth. The X_G is aligned with local vertical (defined from the center of the Earth, parallel to the gravity direction at the launch site). The Z_G is the downrange along the launch azimuth direction and the Y_G completes the right-hand system. The launch azimuth A_z for an ascending orbit is defined by:

$$A_z = \sin^{-1}\left(\frac{\cos i}{\cos \Phi_c}\right) \tag{3.9}$$

in the case of descending orbit,

$$A_z = \frac{\pi}{2} + \sin^{-1}\left(\frac{\cos i}{\cos \Phi_c}\right) \tag{3.10}$$

where i is the inclination of the target orbit. Φ_c is the geocentric latitude at the launch site.

3.3.3 Vehicle Body Reference Frame $X_B Y_B Z_B$

The body axis system is fixed to the launch vehicle as shown in Figure 3.2. The $(OX_B Z_B)$ plane defines the plane of symmetry of the launch vehicle, where X_B axis is coincident with the vehicle body longitudinal axis and the Z_B axis is directed "downwards". The Y_B axis is determined by the right-hand rule. The origin of the axis is fixed at a convenient reference point which is usually, but not necessarily, coincident with the vehicle's center of gravity. The Euler angles (yaw θ, pitch ψ and roll ϕ) are also shown in Figure 3.2.

In this dissertation, the definition of vehicle body axis frame follows the zero-sideslip formulation. The launch vehicle's symmetric plane is assumed to be always the plane formed by the unit vector $\mathbf{1}_b$ along X_B axis and the vehicle's air-relative velocity vector $\mathbf{V}_r$. Thus the sideslip angle keeps zero.

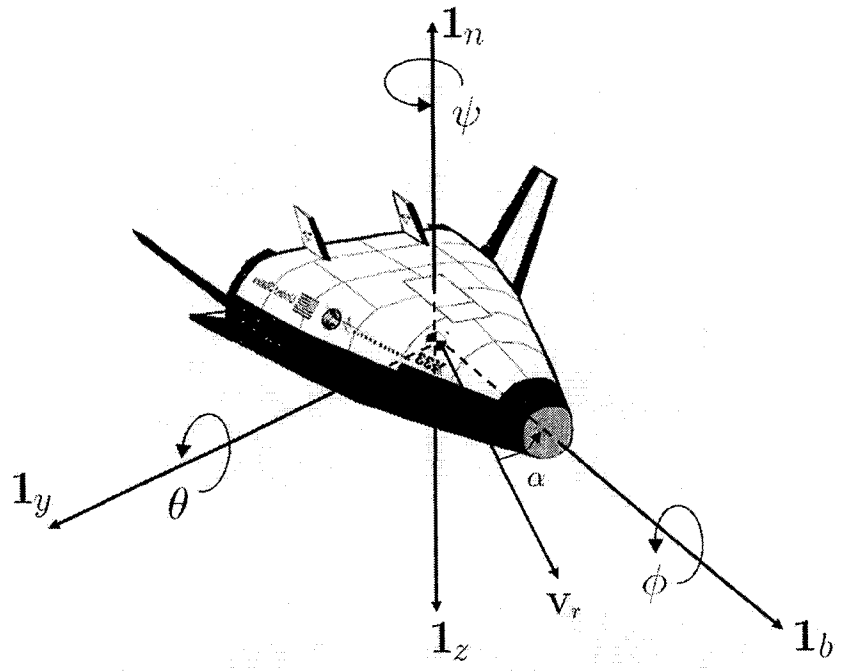

Figure 3.2 Launch vehicle body frame and the relative velocity vector

For the zero-sideslip formulation, the unit vector 1_y along the Y_B axis is given by:

$$\mathbf{1}_y = \frac{\mathbf{V}_r \times \mathbf{1}_b}{\|\mathbf{V}_r \times \mathbf{1}_b\|} \tag{3.11}$$

The unit vector along Z_B follows the right-hand rule:

$$\mathbf{1}_z = \mathbf{1}_b \times \mathbf{1}_y \tag{3.12}$$

Define the body-normal unit vector $\mathbf{1}_n = -\mathbf{1}_z$, thus

$$\mathbf{1}_n = \mathbf{1}_b \times \frac{(\mathbf{1}_b \times \mathbf{V}_r)}{\|\mathbf{1}_b \times \mathbf{V}_r\|} \qquad (\ \alpha > 0\ ) \tag{3.13}$$

From the definition of the dot product and cross product, we have

$$\cos\alpha = \mathbf{1}_b^T \mathbf{1}_{V_r} \quad or \quad |\sin\alpha| = \|\mathbf{1}_{V_r} \times \mathbf{1}_b\| \tag{3.14}$$

where, $\mathbf{1}_{V_r}$ is the unit vector in the direction of $\mathbf{V}_r$. To keep $\mathbf{1}_n$ to be "upwards" constantly, in the case of $\alpha < 0$ the $\mathbf{1}_n$ should be defined as:

$$\mathbf{1}_n = \mathbf{1}_b \times \frac{(\mathbf{V}_r \times \mathbf{1}_b)}{\|\mathbf{1}_b \times \mathbf{V}_r\|} \qquad (\ \alpha < 0\ ) \tag{3.15}$$

Clearly the 1_y should be expressed more reasonably as:

$$\mathbf{1}_y = \mathbf{1}_{V_r} \times \mathbf{1}_b / \sin\alpha \tag{3.16}$$

The above definition is valid for either $\alpha > 0$ or $\alpha < 0$ without causing the instantaneous 180-degree rotation in $\mathbf{1}_y$, when $\mathbf{1}_{V_r}$ and $\mathbf{1}_b$ cross over each other (α changes sign) [41].

3.3.4 Coordinate Transformations

It is frequently necessary to transform motion variables and other parameters from one reference frame to another. Following is a brief review for coordinate transformations in the field of ascent guidance.

Let T_{EG} be the required transformation matrix from ECI reference frame to the guidance frame, thus

$$T_{EG} = \begin{bmatrix} \cos\Theta\cos\Phi_c & \sin\Theta\cos\Phi_c & \sin\Phi_c \\ -\sin\Theta\cos A_z + \cos\Theta\sin\Phi_c\sin A_z & \cos\Theta\cos A_z + \sin\Theta\sin\Phi_c\sin A_z & -\cos\Phi_c\sin A_z \\ -\sin\Theta\sin A_z - \cos\Theta\sin\Phi_c\cos A_z & \cos\Theta\sin A_z - \sin\Theta\sin\Phi_c\cos A_z & \cos\Phi_c\cos A_z \end{bmatrix} \tag{3.17}$$

where Θ and Φ_c are longitude and geodetic latitude of the launch site, respectively. A_z is launch azimuth defined in Eq. (3.9) and Eq. (3.10).

With the rotation sequence of pitch-yaw-roll , the coordinate transformation matrix T_{BG}, which is from body frame to guidance frame, will be

$$T_{BG} = \begin{bmatrix} \cos\theta\cos\psi & \sin\theta\sin\phi - \cos\theta\sin\psi\cos\phi & \sin\theta\cos\phi + \cos\theta\sin\psi\sin\phi \\ \sin\psi & \cos\psi\cos\phi & -\cos\psi\sin\phi \\ -\sin\theta\cos\psi & \cos\theta\sin\phi + \sin\theta\sin\psi\cos\phi & \cos\theta\cos\phi - \sin\theta\sin\psi\sin\phi \end{bmatrix} \tag{3.18}$$

where θ, ψ and ϕ are pitch, yaw, and roll angles respectively.

3.3.5 Determination of Pitch, Yaw, and Roll Angles

In guidance frame, the three unit vectors of the body axes $\mathbf{1}_b$, $\mathbf{1}_y$ and $\mathbf{1}_z$ are indeed the three columns of the transform matrix T_{BG}. Therefore, $\mathbf{1}_b$ is expressed in the guidance frame as:

$$(\mathbf{1}_b)^G = \begin{bmatrix} \cos\theta\cos\psi \\ \sin\psi \\ -\sin\theta\cos\psi \end{bmatrix} \tag{3.19}$$

The unit vector $\mathbf{1}_y$ is expressed in guidance frame as:

$$(\mathbf{1}_y)^G = \begin{bmatrix} \sin\theta\sin\phi - \cos\theta\sin\psi\cos\phi \\ \cos\psi\cos\phi \\ \cos\theta\sin\phi + \sin\theta\sin\psi\cos\phi \end{bmatrix} \tag{3.20}$$

With right-hand rule,

$$(\mathbf{1}_z)^G = (\mathbf{1}_b)^G \times (\mathbf{1}_y)^G \tag{3.21}$$

Once the three unit vectors along the body axes are defined in the guidance frame, the Euler angles can be easily determined by:

$$\begin{aligned} \theta &= -\tan^{-1}\left(\frac{(\mathbf{1}_b)^G{}_z}{(\mathbf{1}_b)^G{}_x}\right) \\ \psi &= \tan^{-1}\left(\frac{(\mathbf{1}_b)^G{}_y}{(\mathbf{1}_b)^G{}_x\cos\theta - (\mathbf{1}_b)^G{}_z\sin\theta}\right) \\ \phi &= -\tan^{-1}\left(\frac{(\mathbf{1}_z)^G{}_y}{(\mathbf{1}_y)^G{}_y}\right) \end{aligned} \tag{3.22}$$

where the superscript "G" means being referred to the guidance frame, while the subscript x, y and z represents the "x", "y" and "z" component respectively.

3.4 Optimization Problem

3.4.1 Performance Index

For a general ascent guidance problem, the optimal control theory is adopted to find the desired thrust direction. Therefore, during endo-atmospheric ascent the vehicle's body-axis orientation $\mathbf{1}_b(t)$, which determines the thrust direction, is optimized at each instant to maximize the performance index.

In this study, the complete ascent trajectory is divided into two portions and the optimal trajectory of each portion is found by the different algorithm. The performance index, which is to minimize the propellant usage, is applied for both of the portions. It can be written as:

$$J = -\int_0^{\tau_f} m' d\tau \tag{3.23}$$

where τ_f is the time-to-go of the complete ascent. m' is the mass flow rate.

3.4.2 Hamiltonian Function and the Costate Differential Equations

The vehicle's mass $m(t)$ is treated as a prescribed function of time in this study. The unit vector $\mathbf{1}_b$ along vehicle's longitudinal body axis is the only control variable. With these assumptions, the Hamiltonian function is defined as:

$$H = \mathbf{p}_r^T \mathbf{V} + \mathbf{p}_v^T \left[-\frac{1}{r^3}\mathbf{r} + (T - A)\mathbf{1}_b + N\mathbf{1}_n \right] - m' + \mu(\mathbf{1}_b{}^T \mathbf{1}_b - 1) \tag{3.24}$$

where $\mathbf{p}_r$ and $\mathbf{p}_v \in R^3$ are the so called costate vectors. μ is a scalar constraint multiplier. The necessary conditions of the optimal control problem call for

$$\begin{cases} \dot{\mathbf{p}}_r = -\dfrac{\partial H}{\partial \mathbf{r}} \\ \dot{\mathbf{p}}_v = -\dfrac{\partial H}{\partial \mathbf{V}} \end{cases} \tag{3.25}$$

More explicitly, we have

$$\begin{cases} \dot{\mathbf{p}}_r = \dfrac{1}{r^3}\mathbf{p}_v - \dfrac{3(\mathbf{p}_v{}^T\mathbf{r})}{r^5}\mathbf{r} - (\mathbf{p}_v{}^T\mathbf{1}_b)\left(\dfrac{\partial T}{\partial \mathbf{r}} - \dfrac{\partial A}{\partial \mathbf{r}}\right) - (\mathbf{p}_v{}^T\mathbf{1}_n)\dfrac{\partial N}{\partial \mathbf{r}} - N\left(\dfrac{\partial \mathbf{1}_n}{\partial \mathbf{r}}\right)^T \mathbf{p}_v \\ \dot{\mathbf{p}}_v = -\mathbf{p}_r + (\mathbf{p}_v{}^T\mathbf{1}_b)\dfrac{\partial A}{\partial \mathbf{V}} - (\mathbf{p}_v{}^T\mathbf{1}_n)\dfrac{\partial N}{\partial \mathbf{V}} - N\left(\dfrac{\partial \mathbf{1}_n}{\partial \mathbf{V}}\right)^T \mathbf{p}_v \end{cases} \tag{3.26}$$

Finally, the costate differential equations can be expressed as [41]:

$$\begin{cases} \dot{\mathbf{p}}_r = \frac{1}{r^3}\mathbf{p}_v - \left[\frac{3(\mathbf{p}_v{}^T\mathbf{r})}{r^4} + (\mathbf{p}_v{}^T\mathbf{1}_b)\left(\frac{\partial T}{\partial r} - A_{\rho r} + \frac{1}{2}C_\rho V_r M_a^2 C_{AM_a}\frac{\partial V_s}{\partial r}\right)\right. \\ \qquad \left. + (\mathbf{p}_v{}^T\mathbf{1}_n)\left(N_{\rho r} - \frac{1}{2}C_\rho V_r M_a^2 C_{NM_a}\frac{\partial V_s}{\partial r}\right)\right]\frac{\mathbf{r}}{r} \\ \qquad + C_\rho \bar{\omega}_E \times \left\{(\mathbf{p}_v{}^T\mathbf{1}_b)\left[(C_A + \frac{1}{2}M_a C_{AM_a})\mathbf{V}_r + \frac{1}{2}C_{A\alpha}V_r^2\frac{\partial \alpha}{\partial \mathbf{V}}\right]\right. \\ \qquad \left. - (\mathbf{p}_v{}^T\mathbf{1}_n)\left[(C_N + \frac{1}{2}M_a C_{NM_a})\mathbf{V}_r + \frac{1}{2}C_{N\alpha}V_r^2\frac{\partial \alpha}{\partial \mathbf{V}}\right]\right\} \\ \dot{\mathbf{p}}_v = -\mathbf{p}_r + C_\rho V_r\left[(\mathbf{p}_v{}^T\mathbf{1}_b)\left(C_A + \frac{1}{2}M_a C_{AM_a}\right) - (\mathbf{p}_v{}^T\mathbf{1}_n)\left(C_A + \frac{1}{2}M_a C_{AM_a}\right)\right]\frac{\mathbf{V}}{V} \\ \qquad + \frac{1}{2}C_\rho V_r^2\left[(\mathbf{p}_v{}^T\mathbf{1}_b)C_{A\alpha} - (\mathbf{p}_v{}^T\mathbf{1}_n)C_{N\alpha}\right] \end{cases} \tag{3.27}$$

where, $A_{\rho r}$, $N_{\rho r}$, C_ρ, C_{AM_a}, C_{NM_a}, $C_{A\alpha}$, $C_{N\alpha}$ are defined as

$$\begin{aligned} A_{\rho r} &= \frac{1}{2m}R_0 V_r^2 S_{ref} C_A \frac{\partial \rho}{\partial r} \\ N_{\rho r} &= \frac{1}{2m}R_0 V_r^2 S_{ref} C_N \frac{\partial \rho}{\partial r} \\ C_\rho &= \frac{1}{m}R_0 \rho S_{ref} \\ C_{AM_a} &= \frac{\partial C_A}{\partial M_a}, \quad C_{NM_a} = \frac{\partial C_N}{\partial M_a} \\ C_{A\alpha} &= \frac{\partial C_A}{\partial \alpha}, \quad C_{N\alpha} = \frac{\partial C_N}{\partial \alpha} \end{aligned}$$

3.4.3 Optimality Condition

$\mathbf{1}_b$ is the only control variable in our trajectory optimization problem formulation. From the standard first order necessary condition, we have

$$\frac{\partial H}{\partial \mathbf{1}_b} = 0 \tag{3.28}$$

It has been shown in the Ref. [7] that the optimal body axis $\mathbf{1}_b{}^*$ lies in the plane formed by the primer vector $\mathbf{p}_v$ and the relative velocity vector $\mathbf{V}_r$. Therefore the search for the optimal body axis orientation can be reduced to a one-dimensional search in the plane of $\mathbf{p}_v$ and $\mathbf{V}_r$. From Figure 3.3, at each instant with given state and costate, we

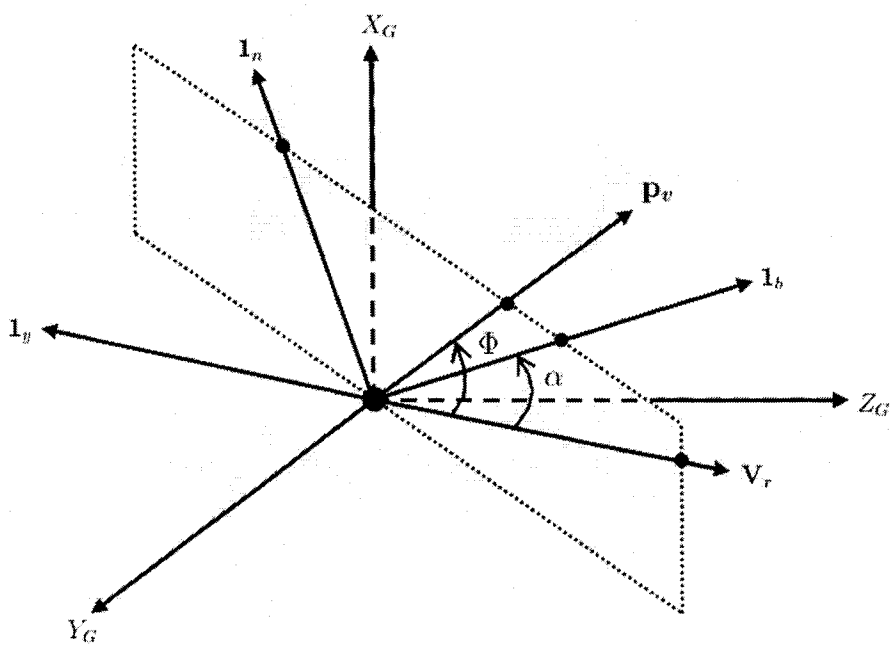

Figure 3.3 Demonstration of the body axes $\mathbf{1}_b\mathbf{1}_y\mathbf{1}_n$, primer vector $\mathbf{p}_v$ and relative velocity vector $\mathbf{V}_r$ in guidance frame $X_GY_GZ_G$

have

$$\cos\Phi = {\mathbf{1}_{P_v}}^T \mathbf{1}_{V_r} \tag{3.29}$$

and,

$$\begin{aligned} {\mathbf{1}_b}^T \mathbf{p}_v &= P_v \cos(\Phi - \alpha) \\ {\mathbf{1}_n}^T \mathbf{p}_v &= P_v \sin(\Phi - \alpha) \end{aligned} \tag{3.30}$$

From above equations, it is clear that maximizing Hamiltonian with respect to $\mathbf{1}_b$ is euivalent to $\frac{\partial H}{\partial \alpha} = 0$, which turns out:

$$\tan(\Phi - \alpha)(T - A + N_\alpha) - (A_\alpha + N) = 0 \tag{3.31}$$

where $N_\alpha = \frac{\partial N}{\partial \alpha}$, $A_\alpha = \frac{\partial A}{\partial \alpha}$. The solution α of the Eq. (3.31) can be found with numerical approach. Once α is obtained, $\mathbf{1}_b{}^*$ can be found with [7]:

$$\mathbf{1}_b{}^* = \frac{\sin\alpha}{\sin\Phi}\mathbf{1}_{P_v} + \frac{\sin(\Phi-\alpha)}{\sin\Phi}\mathbf{1}_{V_r} \tag{3.32}$$

Therefore the optimal $\mathbf{1}_n{}^*$ is:

$$\mathbf{1}_n{}^* = \mathbf{1}_b{}^* \times \frac{\mathbf{1}_b{}^* \times \mathbf{V}_r}{V_r \sin\alpha} \tag{3.33}$$

For the vacuum flight, the aerodynamic terms diminish and α approaches Φ. The optimal body x-axis orientation, therefore the optimal thrust vector, becomes aligned with the costate vector $\mathbf{p}_v$. It is a well-known conclusion in optimal rocket flight in vacuum[25].

3.5 Path Constraints

For safty and vehicle's structural integrity, two kind of most common path constraints are considered in this dissertation: axial acceleration constraint and dynamic pressure constraint.

$$T \leq T_{max} \tag{3.34}$$

$$q \leq q_{max} \tag{3.35}$$

$$\tag{3.36}$$

3.5.1 Axial Thrust Acceleration Constraint

In our study the engine throttle is regarded as a prescribed input. Therefore whether or not and when the axial acceleration constraint will become active are uniquely determined by the prescribed engine throttle. Once this constraint is activated, the throttle will be adjusted with

$$\eta = \frac{T_{max} m(t) g_0}{T_{vac}} \tag{3.37}$$

where T_{max} is the maximum allowable value of the axial thrust acceleration which is a design parameter of the launch vehicle.

3.5.2 Dynamic Pressure Constraint

The dynamic pressure is defined as:

$$q = \frac{1}{2}\rho V^2 \tag{3.38}$$

The velocity of a rocket powered launch vehicle during launch constantly increases with the increment of altitude. Therefore, the dynamic pressure on a rocket power launch vehicle is initially zero because the velocity is zero. The dynamic pressure increases because of the increasing velocity to some maximum value, called the maximum dynamic pressure, or q_{max}. Thereafter the dynamic pressure decreases because of the decreasing density. The q_{max} condition is a design constraint on full scale rocket powered launch vehicle.

Ref. [7] presents an effective approach to impose the dynamic pressure constraint through adjusting engine throttle. It is briefly reviewed in following part.

Let $\delta > 0$ be a small time increment. A first-order approximation of $q(t+\delta)$ is

$$\begin{aligned} q(t+\delta) &\approx q(t) + \dot{q}(t)\delta \\ &= q(t) + [c_1 + c_2\eta(t)]\delta \end{aligned} \tag{3.39}$$

where $\eta(t)$ is the current engine throttle. c_1 and c_2 are respectively defined as [7]:

$$\begin{aligned} c_1 =& \frac{1}{2}V_r^2\rho_r\frac{\mathbf{r}^T\mathbf{V}}{r} - \rho\left[\frac{\mathbf{r}^T\mathbf{V}_r}{r^3} + AV_r\cos\alpha + NV_r\sin\alpha\right] \\ c_2 =& \rho V_r T\cos\alpha \end{aligned} \tag{3.40}$$

Let $q(t+\delta) \leq q_{max}$. From Eq. 3.39 and Eq. 3.40, we have

$$\eta(t) \leq \frac{q_{max} - q(t) - c_1\delta}{c_2\delta} \triangleq \eta_q \tag{3.41}$$

Hence the current engine throttle command η can be determined with:

$$\eta = \begin{cases} \eta_0, & if \quad \eta_q > \eta_0 \\ \eta_q, & if \quad \eta_{min} \leq \eta_q \leq \eta_0 \\ \eta_{min}, & if \quad \eta_q < \eta_{min} \end{cases} \tag{3.42}$$

where η_0 is the prescribed current engine throttle; η_{min} is the minimum allowable engine throttle setting during engine-on period. The last case in Equation 3.42 is the one where the dynamic pressure constraint can not be met through by lowering the engine throttle within the allowable range. It is an unlikely event in real flight of the launch vehicle.

CHAPTER 4. EXO-ATMOSPHERIC ASCENT GUIDANCE FORMULATION

Once the vehicle reaches an altitude where the atmospheric density is sufficiently low, the aerodynamic forces and the wind can be neglected from the dynamics model of the vehicle. For this stage, a semi-analytical solution is available for solving the state and co-state variables along the optimal trajectory. It makes the searching process for vehicle's optimal vacuum trajectory to be very reliable and fast.

Three assumptions are accepted for followed discussion:

- It is the vacuum flight where the only non-propulsion force is the gravity.
- The magnitude of thrust is a fixed function of time.
- The thrust vector is aligned with the longitudinal body axis.

4.1 Vacuum Equations of Motion of Launch Vehicles

Similar with the case of atmospheric flight in Chapter 3, the dynamics of a launch vehicle in vacuum flight is modeled in a Newtonian central gravitational field. Being referred to an inertial coordinate system, the equations of motion for a rocket-powered

launch vehicle in vacuum flight can be written as:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{V} \\ \dot{\mathbf{V}} = \mathbf{g}(\mathbf{r}) + + \dfrac{T\mathbf{1}_b}{m(t)} \\ \dot{m} = -\dfrac{\eta T_{vac}}{g_0 I_{sp}} \end{cases} \tag{4.1}$$

where $\mathbf{r}$ and $\mathbf{V} \in R^3$ are the position and inertial velocity vectors respectively; $\mathbf{g}$ is the gravitational acceleration which is a function of $\mathbf{r}$; T_{vac} is the magnitude of full vacuum thrust; $\eta > 0$ is the engine throttle; and T is the current thrust magnitude including effect of throttle modulation. In this formulation, the total engine thrust is assumed to be aligned with the body longitudinal axis $\mathbf{1}_b$; I_{sp} is the specific impulse of the engine and g_0 represents the magnitude of the gravitational acceleration at the radius r_0. An ingenious approximation to the gravitational acceleration is the so-called linear gravity approximation

$$\mathbf{g} = -\frac{\mu}{\bar{r}^2}\frac{\mathbf{r}}{\bar{r}} = -\bar{\omega}^2\mathbf{r} \tag{4.2}$$

where μ is the gravitational parameter of the Earth, $\bar{r}$ is another reference radius (e.g., an average value of r along the ascent trajectory), and $\bar{\omega} = \sqrt{\mu/\bar{r}^3}$ is the Schuler frequency at $\bar{r}$. This approximation preserves the more important characteristic of the gravity in ascent flight, the direction, and enables analytical solution to the co-state equation in optimal vacuum flight as shall be seen later. The minor gravity magnitude difference caused by this approximation in ascent typically has little influence on the precision of the trajectory.

For better numerical conditioning, the distances are normalized by r_0, the velocities by $\sqrt{r_0 g_0}$, and the time by $\sqrt{r_0/g_0}$. With some abuse of notation, we will still use $\mathbf{r}$ and $\mathbf{V}$ to denote the dimensionless position and velocity vector, respectively. With the above linear gravity approximation, the dimensionless equations of motion are:

$$\begin{cases} \mathbf{r}' = \mathbf{V} \\ \mathbf{V}' = -\omega^2 \mathbf{r} + A_T \mathbf{1}_T \end{cases} \tag{4.3}$$

where the differentiation is with respect to the nondimensional time $\tau = t/\sqrt{r_0/g_0}$, $\omega = \sqrt{(r_0/\bar{r})^3}$ is the nondimensional Schuler frequency, and $A_T = T/mg_0$ is the instantaneous thrust acceleration in g. The vehicle's mass flow rate equation becomes

$$m' = -\frac{T}{c} \tag{4.4}$$

where $c = I_{sp}/\sqrt{r_0/g_0^3}$ is a piecewise constant.

4.2 Optimization Problem

As mentioned in Chapter 3, the multiple-burn sequence is introduced to include an optimal coast in between the two vacuum burns for the exo-atmospheric ascent. The main purpose is to be more efficient on fuel usage. Therefore the objective of the optimization problem is to minimize the fuel usage or, equivalently, maximize the vehicle mass at orbital insertion.

4.2.1 Performance Index

The guidance command for $\mathbf{1}_T$ is determined by the solution of an optimal control problem in which the performance index is defined as the minimization of fuel usage (See Eq. 3.23). As mentioned in Chapter 2, the burn time of the endo-atmospheric ascent portion is fixed by the carried propellant and the mass flow rate of the engine. In other words, the propellant usage is fixed for the endo-atmospheric ascent portion. Therefore the performance index Eq. 3.23 can be equivalently expressed as:

$$J = -\int_{\tau_1}^{\tau_f} m' d\tau \tag{4.5}$$

where τ_1 is the instant at which the atmospheric ascent ends and the vacuum ascent starts. m' is the mass flow rate as mentioned before. Therefore $\int_{\tau_1}^{\tau_f} m' d\tau$ is the total fule usage during exo-atmospheric ascent portion $[\tau_1, \tau_f]$.

4.2.2 Hamiltonian Function and the Costate Differential Equations

The standard optimal control theory[30] calls for the use of the Hamiltonian

$$H = \mathbf{p}_r^T \mathbf{V} - \omega^2 \mathbf{p}_V^T \mathbf{r} + \mathbf{p}_V^T \mathbf{1}_T A_T - p_m \frac{T}{c} - \frac{T}{c}$$

Re-write the Hamiltonian as:

$$H = \mathbf{p}_r^T \mathbf{V} - \omega^2 \mathbf{p}_V^T \mathbf{r} + T \left(\frac{\mathbf{p}_V^T \mathbf{1}_T}{m g_0} - \frac{p_m}{c} - \frac{1}{c} \right) := H_{NT} + TS \tag{4.6}$$

where $\mathbf{p}_r$ and $\mathbf{p}_V$ are the costate vectors corresponding to position and inertial velocity vectors respectively. $\mathbf{p}_m$ is the costate variable which is corresponding to the state variable m. From necessary conditions of the optimal control problem, the costate vectors will satisfy the differential equation

$$\begin{pmatrix} \mathbf{p}_r' \\ \mathbf{p}_V' \end{pmatrix} = - \begin{pmatrix} \frac{\partial H}{\partial \mathbf{r}} \\ \frac{\partial H}{\partial \mathbf{V}} \end{pmatrix} = \begin{pmatrix} \omega^2 \mathbf{p}_V \\ -\mathbf{p}_r \end{pmatrix} \tag{4.7}$$

In particular $\mathbf{p}_V$ is called the primer vector because the optimal thrust direction $\mathbf{1}_T = \mathbf{p}_V / ||\mathbf{p}_V||$.[25] The switching function S determines when to use full thrust and when to coast ($T = 0$). Specifically,

$$T = \begin{cases} T_{max} & \text{if } S > 0, \\ 0 & \text{if } S < 0. \end{cases} \tag{4.8}$$

The case of singular thrust arcs (when $S \equiv 0$ in a finite period of time) is not considered because it is well known that, except for a few pathological cases, they are not optimal in vacuum flight.[43] Suppose in our case that the coast between the 2nd and 3rd stage ends at a to-be-determined time τ_{OMS}. It is then necessary that $S(\tau_{OMS}) = 0$ (and

$S(\tau) > 0$ for $\tau > \tau_{OMS}$.) Since the total final time τ_f is free for the optimal ascent problem, the Hamiltonian H is equal to zero along the optimal trajectory.[30]

$$H(t) = 0, \qquad (\ t_0 \leq t \leq t_f\ ) \tag{4.9}$$

As a result, we have equivalently

$$H_{NT}(\tau_{OMS}) = 0 \tag{4.10}$$

This equation avoids the computation of the mass costate p_m thus is preferred to the condition $S(\tau_{OMS}) = 0$. For multiple coast arcs, refer to Ref. [23] for additional interior point conditions.

4.2.3 Terminal Conditions

For optimal ascent guidance problem, the terminal conditions include the final orbital insertion conditions and transversality conditions. The orbital insertion conditions typically specify some of the 6 target orbital elements. In general, they can be written as k $(0 < k \leq 6)$ algebraic conditions:

$$\Psi(\mathbf{r}(t_f), \mathbf{V}(t_f)) = 0, \quad \Psi \in R^k \quad (0 < k \leq 6) \tag{4.11}$$

For a general optimal control problem, the performance index is:

$$J = \phi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}, u, t)dt \tag{4.12}$$

where $\mathbf{x} = col(\mathbf{r}\ \mathbf{V})$; $\mathbf{x}_f$ is the final state variable at t_f; $\mathbf{x}$ and u are state and control variables respectively. Therefore we have following transversality conditions:

$$\mathbf{p}_r(t_f) = -\frac{\partial\phi(\mathbf{r}_f, \mathbf{V}_f, t_f)}{\partial\mathbf{r}_f} + \left(\frac{\partial\Psi}{\partial\mathbf{r}_f}\right)^T \bar{\nu} \tag{4.13}$$

$$\mathbf{p}_v(t_f) = -\frac{\partial\phi(\mathbf{r}_f, \mathbf{V}_f, t_f)}{\partial\mathbf{V}_f} + \left(\frac{\partial\Psi}{\partial\mathbf{V}_f}\right)^T \bar{\nu} \tag{4.14}$$

where $\bar{\nu} \in R^k$ is a constant multiplier vector. The 6 transversality conditions can be combined into $6 - k$ independent conditions by eliminating $\bar{\nu}$. Two examples will be shown later for eliminating $\bar{\nu}$ to get independent conditions.

4.2.4 Examples of Different Terminal Conditions

Example 1: Four Insertion Constraints Case

Let a^*, e^*, i^* be the semi-major axis, eccentricity and inclination of target orbit respectively. γ_f^* is the vehicle's flight path angle at orbital insertion point and which is zero.

Therefore the magnitude of the position and velocity vectors at orbital insertion will be:

$$V_f^* = \sqrt{\frac{1.0 + e^*}{a^*(1.0 - e^*)}} \tag{4.15}$$

$$r_f^* = a^*(1.0 - e^*) \tag{4.16}$$

which is for the case of insertion at perigee.

$$V_f^* = \sqrt{\frac{1.0 - e^*}{a^*(1.0 + e^*)}} \tag{4.17}$$

$$r_f^* = a^*(1.0 + e^*) \tag{4.18}$$

which is for the case of insertion at apogee.

Let us define the unit vector $\mathbf{1}_{H0}$ in Earth Center Inertial frame (See Section 3.3.4):

$$\mathbf{1}_{H0} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4.19}$$

In the Guidance Frame (See Section 3.3.5), we have

$$\mathbf{1}_{L0} = T_{EP} \cdot \mathbf{1}_{H0} \tag{4.20}$$

Therefore the four orbital insertion conditions $\Psi \in R^4$ can be written as:

$$\frac{1}{2}\mathbf{r}_f{}^T \cdot \mathbf{r}_f - \frac{1}{2}r_f^{*2} = 0 \tag{4.21}$$

$$\frac{1}{2}\mathbf{V}_f{}^T \cdot \mathbf{V}_f - \frac{1}{2}V_f^{*2} = 0 \tag{4.22}$$

$$\mathbf{1}_{L0}{}^T \cdot (\mathbf{r}_f \times \mathbf{V}_f) - r_f^* V_f^* \cos i^* = 0 \tag{4.23}$$

$$\mathbf{r}_f{}^T \cdot \mathbf{V}_f = 0 \tag{4.24}$$

Substitute above Ψ into the transversality conditions 4.13 and 4.14. After eliminating the multiplier vector $\bar{\nu} \in R^4$, two transversality conditions will be:

$$\mathbf{V}_f{}^T \cdot \mathbf{p}_{r_f} r_f^{*2} - \mathbf{r}_f{}^T \cdot \mathbf{p}_{V_f} V_f^{*2} = 0 \tag{4.25}$$

$$\begin{aligned}&[(\mathbf{r}_f \times \mathbf{V}_f)^T \cdot \mathbf{p}_{r_f}][(\mathbf{r}_f \times \mathbf{V}_f)^T \cdot (\mathbf{r}_f \times \mathbf{1}_{L0})] \\ &+ [(\mathbf{r}_f \times \mathbf{V}_f)^T \cdot \mathbf{p}_{V_f}][(\mathbf{r}_f \times \mathbf{V}_f)^T \cdot (\mathbf{V}_f \times \mathbf{1}_{L0})] = 0\end{aligned} \tag{4.26}$$

The above 2 transversality conditions plus the 4 orbital insertion conditions 4.21-4.24 constitute the 6 terminal conditions for this case.

Example 2: Five Insertion Constraints Case

Let r_f^* and V_f^* are given magnitudes of the position and velocity vectors at orbital insertion. $\gamma_f^* = 0$ is the given vehicle's flight path angle at orbital insertion. i^* and Ω^* are desired orbital inclination and longitude of ascending node.

Let us define the unit vector $\mathbf{1}_{H1}$ in the Earth Center Inertial frame (See Section 3.3.4). It is along the direction of angular momentum.

$$\mathbf{1}_{H1} = \begin{bmatrix} \sin\Omega^* \sin i^* \\ -\cos\Omega^* \sin i^* \\ \cos i^* \end{bmatrix} \tag{4.27}$$

In the Guidance Frame (See Section 3.3.5), we have

$$\mathbf{1}_{L1} = T_{EP} \cdot \mathbf{1}_{H1} \tag{4.28}$$

Hence the five orbital insertion conditions $\Psi \in R^5$ can be written as:

$$\frac{1}{2}\mathbf{r}_f{}^T \cdot \mathbf{r}_f - \frac{1}{2}r_f^{*2} = 0 \tag{4.29}$$

$$\frac{1}{2}\mathbf{V}_f{}^T \cdot \mathbf{V}_f - \frac{1}{2}V_f^{*2} = 0 \tag{4.30}$$

$$\mathbf{r}_f{}^T \cdot \mathbf{V}_f = 0 \tag{4.31}$$

$$\mathbf{r}_f^T \mathbf{1}_{L1} = 0 \tag{4.32}$$

$$\mathbf{V}_f^T \mathbf{1}_{L1} = 0 \tag{4.33}$$

Substitute above five orbital insertion conditions Ψ into the transversality conditions 4.13 and 4.14. After eliminating the multiplier vector $\bar{\nu} \in R^5$, the transversality conditions will be:

$$(\mathbf{V}_f \cdot \boldsymbol{P}_{r_f})r_f^{*2} - (\mathbf{r}_f^T \cdot \boldsymbol{P}_{V_f})V_f^{*2} = 0 \tag{4.34}$$

4.3 Path Constraint

4.3.1 Axial Thrust Acceleration Constraint

In the vacuum flight, only one path constraint is considered in ascent guidance: vehicle's axial thrust acceleration constraint.

$$A_T \leq T_{max} \tag{4.35}$$

Once this constraint is activated, the prescribed engine throttle will be adjusted as:

$$\eta = \frac{T_{max} m(t) g_0}{T_{vac}} \tag{4.36}$$

where T_{max} is the maximum allowable value of the axial thrust acceleration which is a design parameter of the launch vehicle.

4.3.2 Constant Acceleration Arc

For the "second powered stage" (See Section 2.2) in exo-atmospheric ascent portion, the time at which Main Engine Cut-Off (MECO) is determined by the carried propulsion

propellant and the mass flow rate of the main engine. Therefore it is a fixed parameter. Let M_{mp} is the carried propulsion propellant of the main engine. The mass flow rate of the main engines is $M_{rate}(kg/sec.)$. Let τ_1 to be the instant at which the endo-atmospheric ascent portion ends and τ_{meco} to be the instant at which the main engine cut off. We have:

$$\tau_{meco} = \tau_1 + \frac{M_{mp}}{M_{rate}} \tag{4.37}$$

However with the axial thrust acceleration constraint, it is highly possible that there is a constant acceleration arc along the ascent trajectory during the "second powered stage". In this case, calculation of the τ_{meco} is a little complicated. The duration $[\tau_1, \tau_{meco}]$ can be divided into two portions: $[\tau_1, \tau_{ca}]$ and $[\tau_{ca}, \tau_{meco}]$. The Axial Acceleration Constraint is activated at τ_{ca}. The trajectory in the period of $[\tau_{ca}, \tau_{meco}]$ is called the Constant Acceleration Arc. During $[\tau_1, \tau_{ca}]$ the engine throttle is set to be 1. Let suppose the mass of the launch vehicle is M_0 at the beginning of exo-atmospheric ascent. T_{me} is the magnitude of thrust of the Main Engine. If T_{max} is the maximum allowable axial thrust acceleration, then

$$\tau_{ca} = \tau_1 + \frac{(M_0 - \frac{T_{me}}{T_{max}})}{M_{rate}} \tag{4.38}$$

If $m(t)$ is the current mass of the vehicle after the axial thrust acceleration constraint being activated, η is the current throttle, and $\tau = t - \tau_{ca}$. We have:

$$T_{max} \equiv \eta \frac{T_{me}}{m(t)} \tag{4.39}$$

where

$$m(t) = M_0 - (\tau_{ca} - \tau_1) M_{rate} + \int_0^{\tau} \eta \cdot M_{rate}\, d\sigma \tag{4.40}$$

$$= M_{ca} + \int_0^{\tau} \eta \cdot M_{rate}\, d\sigma \tag{4.41}$$

where M_{ca} is the vehicle's mass at τ_{ca}. From Eq. 4.39 and Eq. 4.41, we have:

$$\eta T_{me} = T_{max} \cdot M_{ca} + T_{max} \cdot M_{rate} \int_0^{\tau} \eta\, d\sigma \tag{4.42}$$

Differentiation with respect to τ,

$$\dot{\eta} T_{me} = T_{max} \cdot M_{rate} \cdot \eta \tag{4.43}$$

Take the integral for both sides of the above equation, it turns out

$$\eta(\tau) = \eta(\tau_{ca}) e^{\frac{T_{max} \cdot M_{rate}}{T_{me}} \tau} \tag{4.44}$$

$$= e^{\frac{T_{max} \cdot M_{rate}}{T_{me}} \tau} \tag{4.45}$$

Therefore on the constant acceleration arc, the instantaneous engine throttle is

$$\eta(t) = e^{\frac{T_{max} \cdot M_{rate}}{T_{me}} (t - \tau_{ca})}, \quad t \geq \tau_{ca} \tag{4.46}$$

Let $M_{meco} = M_0 - M_{mp}$, in the case of with constant acceleration arc, τ_{meco} is obtained with:

$$\tau_{meco} = \tau_{ca} + \frac{T_{me}}{T_{max} \cdot M_{rate}} \ln \frac{M_{meco}}{M_{ca}} \tag{4.47}$$

CHAPTER 5. NUMERICAL ALGORITHMS

In this dissertation, for algorithmic design the complete ascent trajectory is partitioned into two portions: the atmospheric and vacuum portion. The atmospheric ascent trajectory is from lift-off from launch pad to a sufficiently high altitude where the dynamic pressure is ignorable for guidance purpose. The rest of the trajectory will be considered vacuum. The ascent trajectory will be tacitly assumed to be three powered stages (or alternatively, 3 burns) with a coast between the 2nd and 3rd powered stage. Only the first stage flies the atmospheric ascent trajectory. The burn times of the 1st and 2nd stage are fixed by the propellant carried and mass flow rates. The coast time and the burn time of the 3rd stage are to be optimized. This division of powered stages can be due to the actual separation of the physical stages or launch sequence. In the case of the Space Shuttle, for instance, the first stage will be the entire stack until the separation of the Solid Booster Rockets; the second stage will be the Shuttle until Main Engine Cut-Off (MECO), and the third stage will be the Shuttle powered by the Orbital Maneuver System (OMS) after the coast from MECO. In the case where the physical staging is not as exactly defined as 3 powered stages, the first stage may be artificially considered to be the trajectory within a specified amount of flight time from lift-off. This time is chosen to ensure that at the end of the "first stage", the dynamic pressure is below a threshold value. The "second powered stage" will then be rest of the burn before the coast. The "3rd powered stage" in the last burn that achieves the targeting condition. Even though only one coast arc between the 2nd and 3rd stage is considered in the following, the methodology in this dissertation is readily applicable to the cases

with more powered stages and multiple coasts, provided that the number of coast arcs is specified in advance.

Two main numerical algorithms are presented in this dissertation for rapid planning and closed-loop guidance for ascent through atmosphere. For endo-atmospheric portion, a novel Fixed-Point Iteration (FPI) algorithm is developed for solving a special form of TPBVP formulated through integration of Atmospheric Ascent and Vacuum Ascent. The Analytical Multiple-Shooting Vacuum algorithm is developed to be responsible for searching vacuum optimal trajectories with optimal coast in between the burns. The convergence of the fixed-point iteration is one of the critical concerns for the FPI algorithm. Therefore several numerical techniques are exploited to address this issue.

5.1 Integration of Atmospheric Ascent and Vacuum Ascent

The atmospheric portion of the ascent trajectory subject to a number of commonly seen path constraints is found by using the Finite Difference (FD) algorithm presented in Ref. [7]. In this case the control vector $\mathbf{1}_T$ not only defines the direction of the thrust vector, but affects nonlinearly the magnitude and direction of $\mathbf{A}$ and $\mathbf{N}$ as well (See Chapter 3). In addition, $\mathbf{A}$ and $\mathbf{N}$ are also highly nonlinear functions of the state. The problem now is much more complex and there are no closed-form solutions, exact or approximate, exist for the state and costate. The FD algorithm represents an efficient numerical approach to solve the TPBVP resulting from the optimal atmospheric ascent problem, and more detail on the development and testing of the algorithm can be found therein. To apply the FD algorithm in conjunction with the vacuum algorithm discussed in the preceding sections, the TPBVP and the exchange of data between the two algorithms need to be planned properly so that, when converged, the results from the two algorithms indeed represent different parts of the same optimal trajectory.

Our approach is to integrate the algorithms through the iteration on the state at the

end point τ_1 of the first stage where the atmospheric algorithm stops and the vacuum algorithm begins. From a starting estimate of the state at τ_1, the vacuum algorithm generates an optimal trajectory to the target orbit and returns the corresponding costate at τ_1. The atmospheric algorithm in turn solves a TPBVP in which the initial state is that at the lift-off and the final costate is the one just found by the vacuum algorithm at τ_1. This atmospheric ascent solution will provide a new state at τ_1. The above process then repeats until the states found at τ_1 in two consecutive cycles are practically the same.

The algorithmic implementation of the above approach is given below. Starting from an initial guess of the state at τ_1, denoted by $\mathbf{x}_{\tau_1}^{(0)}$, our integration algorithm proceeds with the iterations as follows:

1. Set $k = 0$.

2. With the known $\mathbf{x}_{\tau_1}^{(k)}$, the vacuum algorithm finds the optimal ascent trajectory in $(\tau_1,\ \tau_f)$, including the coast, from $\mathbf{x}_{\tau_1}^{(k)}$ to the targeting conditions. As a part of the solution the vacuum algorithm returns the corresponding costate $\mathbf{p}_{\tau_1}^{(k)}$ at τ_1.

3. The just obtained $\mathbf{p}_{\tau_1}^{(k)}$ is used as the required boundary condition at τ_1 for the costate in the interval $(0,\ \tau_1)$ for the atmospheric algorithm. The initial state at $\tau = 0$ is known. A special form of the TPBVP for the atmospheric algorithm is then well defined. Upon the convergence of the algorithm, a new state at τ_1 is found as a result, and this new state is denoted as $\tilde{\mathbf{x}}_{\tau_1}^{(k+1)}$.

4. If $||\tilde{\mathbf{x}}_{\tau_1}^{(k+1)} - \mathbf{x}_{\tau_1}^{(k)}|| \leq \delta$ for some pre-selected small constant $\delta > 0$, set $\mathbf{x}_{\tau_1} = \tilde{\mathbf{x}}_{\tau_1}^{(k+1)}$, and stop. Otherwise, let

$$\mathbf{x}_{\tau_1}^{(k+1)} = \varepsilon\tilde{\mathbf{x}}_{\tau_1}^{(k+1)} + (1-\varepsilon)\mathbf{x}_{\tau_1}^{(k)} \tag{5.1}$$

where $0 < \epsilon \leq 1$ is a constant. Set $k = k + 1$ and return to Step 2 above.

At the conclusion of the above process, both the state $\mathbf{x}$ and costate $\mathbf{p}$ are continuous at τ_1. All the necessary conditions for the optimal ascent problem are satisfied in $[0,\ \tau_f]$. In other words, the solutions obtained for $\mathbf{x}(\cdot)$ and $\mathbf{p}(\cdot)$ in the two algorithms form a continuous extremal for the complete optimal ascent problem. The flow chart of integration algorithm is given in Figure 5.1

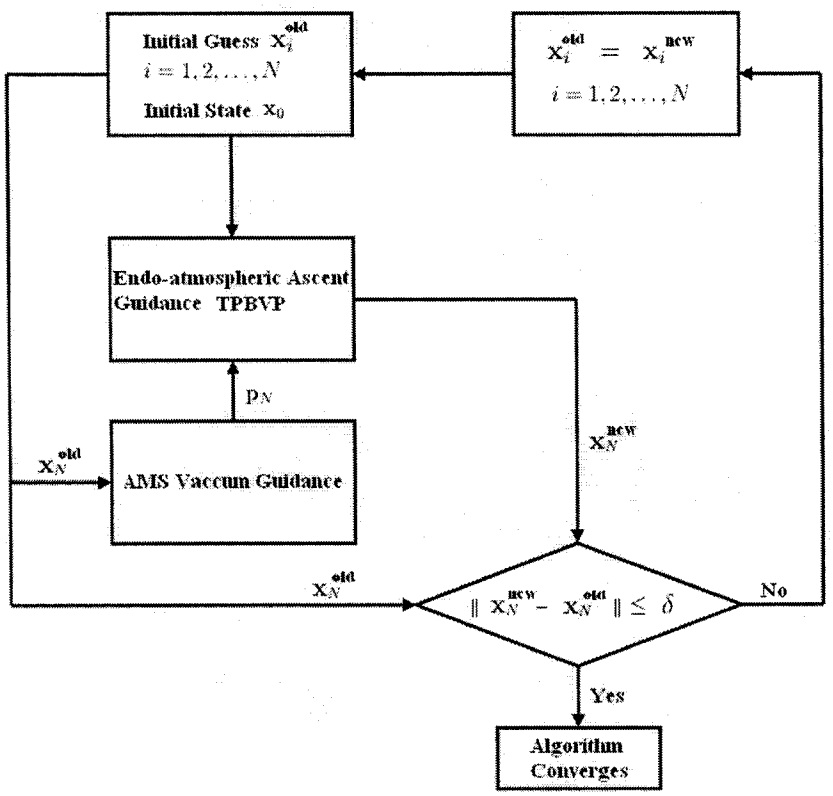

Figure 5.1 Flow chart of the trajectory integration algorithm

5.2 Numerical Algorithms for Endo-atmospheric Ascent

5.2.1 Brief Review of the Finite Difference (FD) algorithm

The algorithm integration approach in Section 5.1 is generic in that it is not dependant on the specific algorithm. The Finite Difference (FD) algorithm in Ref. [7] is well suited for solving the endo-atmospheric ascent guidance problem. Let us briefly review the Finite Difference method for endo-atmospheric ascent.

The optimality condition dictates that the direction of the optimal control $\mathbf{1}_T$ is completely determined by the state $\mathbf{x}$ and the primer vector $\mathbf{p}_V$ (see Ref. [7]). Thus after $\mathbf{1}_T$ is replaced as a function of the state and costate, the differential equation system for the optimal atmospheric ascent problem in $[0,\ \tau_1]$ can be written as

$$\mathbf{y}' = \begin{pmatrix} \mathbf{x}' \\ \mathbf{p}' \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1(\tau, \mathbf{x}, \mathbf{p}) \\ \mathbf{f}_2(\tau, \mathbf{x}, \mathbf{p}) \end{pmatrix} = \mathbf{f}(\tau, \mathbf{y}) \tag{5.2}$$

The state equations $\mathbf{f}_1$ consist the right hand sides of Eq. (3.2). The costate equations $\mathbf{f}_2$ consist the right hand side of Eq. (3.27). Both $\mathbf{f}_1$ and $\mathbf{f}_2$ are continuously differentiable with respect to their arguments. The boundary conditions are

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{5.3}$$

$$\mathbf{p}(\tau_1) = \mathbf{p}_{\tau_1}^{(k)} \tag{5.4}$$

where $\mathbf{x}_0$ is the known initial state at lift-off, and $\mathbf{p}_{\tau_1}^{(k)}$ is the costate returned from the vacuum algorithm in Step 2 of the integration algorithm. The FD algorithm proceeds by dividing the time interval $[0,\ \tau_1]$ into N sub-intervals of the same length $h = \tau_1/N$. Let $\mathbf{y}_i = \mathbf{y}(ih)$ be the value of the solution at the node $\tau_i = ih$, $i = 0, \ldots, N$. At the middle point between τ_{i-1} and τ_i, denoted by $\tau_{i-1/2} = \tau_i - h/2$, the differential equations (5.2) are approximated by central finite difference at $\tau_{i-1/2}$:

$$\frac{1}{h}(\mathbf{y}_i - \mathbf{y}_{i-1}) = \mathbf{f}\left(\tau_{i-1/2}, \frac{\mathbf{y}_i + \mathbf{y}_{i-1}}{2}\right), \quad i = 1, \ldots, N \tag{5.5}$$

The above approximation converts the differential equation (5.2) into equality constraints on $\{\mathbf{y}_i\}$. The search for $\{\mathbf{y}_i\}$ becomes the problem of solving the system nonlinear algebraic equations in Eq. (5.5). Because of a special banded structure of the Jacobian of the problem, a modified Newton method, Gauss elimination and back substitutions can be employed to solve the problem without the need to compute the inverse of the large Jacobian matrix.[7] The solutions to $\{\mathbf{y}_i\}$ so obtained are second-order approximations to the solution of the TPBVP at τ_i.

5.2.2 Fixed-Point Iteration (FPI) Method

The special structure of the TPBVP formulated in Step 3 of the algorithm in Section 5.1 gives rise to an interesting possibility: that the optimal endo-atmospheric ascent problem may be solved by a very simple algorithm based on nested fixed-point iterations. In the current setting of this research, we separate and re-organize the finite-difference equations (5.5) for $\mathbf{x}$ and $\mathbf{p}$ as

$$\mathbf{x}_i = \mathbf{x}_{i-1} + h\mathbf{f}_1\left(\tau_{i-1/2}, \frac{\mathbf{x}_i + \mathbf{x}_{i-1}}{2}, \frac{\mathbf{p}_i + \mathbf{p}_{i-1}}{2}\right), \quad i = 1, \ldots, N \tag{5.6}$$

$$\mathbf{p}_{i-1} = \mathbf{p}_i - h\mathbf{f}_2\left(\tau_{i-1/2}, \frac{\mathbf{x}_i + \mathbf{x}_{i-1}}{2}, \frac{\mathbf{p}_i + \mathbf{p}_{i-1}}{2}\right), \quad i = N, \ldots, 1 \tag{5.7}$$

We make two key observations on the solutions to Eqs. (5.6) and (5.7). Consider first the case when a set of the nodal values of $\{\mathbf{p}_i\}$ are given and $\{\mathbf{x}_i\}$ are to be determined from Eq. (5.6). Notice that the i-th equation in (5.6) involves only $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$. Suppose that for $i = 1$, $\mathbf{x}_0 = \mathbf{x}(0)$ is taken according to the boundary condition (5.3). Then everything on the right hand side of Eq. (5.6) for $i = 1$ is known except for $\mathbf{x}_1$. That is, Eq. (5.6) for $i = 1$ becomes essentially a fixed-point equation in $\mathbf{x}_1$

$$\mathbf{x}_1 = \boldsymbol{\zeta}_1(\mathbf{x}_1) \tag{5.8}$$

where $\boldsymbol{\zeta}_1(\mathbf{x}_1)$ is the right hand side of Eq. (5.6) with $i = 1$. Once the value of $\mathbf{x}_1$ is found from the above equation and substituted into Eq. (5.6) for $i = 2$, the same reasoning

will lead to a similar fixed-point equation in $\mathbf{x}_2$. Repeating this process successively, we conclude that in general we have at the i-th node the fixed-point equation to solve

$$\mathbf{x}_i = \boldsymbol{\zeta}_i(\mathbf{x}_i), \quad i = 1, 2, \ldots, N \tag{5.9}$$

Keep in mind that in above equations each function $\boldsymbol{\zeta}_i$ is dependent on the given $\mathbf{p}_i$, $\mathbf{p}_{i-1}$, and $\mathbf{x}_{i-1}$ found as the solution of the previous equation $\mathbf{x}_{i-1} = \boldsymbol{\zeta}_i(\mathbf{x}_{i-1})$, with $\mathbf{x}_0 = \mathbf{x}(0)$.

Next, consider the case when $\{\mathbf{x}_i\}$ are given and $\{\mathbf{p}_i\}$ are to be found from Eq. (5.7). Suppose that we set $\mathbf{p}_N = \mathbf{p}_{\tau_1}^{(k)}$ as required by the boundary condition (5.4). Application to Eq. (5.7) an argument similar to the above one, but backward in time, will lead to fixed-point equations for $\mathbf{p}_i$s

$$\mathbf{p}_i = \boldsymbol{\eta}_i(\mathbf{p}_i), \quad i = N-1, \ldots, 1, 0 \tag{5.10}$$

Likewise the functions $\boldsymbol{\eta}_i$ is dependent on the given $\mathbf{x}_i$, $\mathbf{x}_{i+1}$, and $\mathbf{p}_{i+1}$ found as the solution of the previous (backward in time) equation $\mathbf{p}_{i+1} = \boldsymbol{\eta}_i(\mathbf{p}_{i+1})$, with $\mathbf{p}_N = \mathbf{p}_{\tau_1}^{(k)}$.

The fixed-point algorithm we propose can have two versions, differing mainly in the order of the fixed-point iterations involved. The reason for the difference is that one version may have better convergence than the other in a particular problem, as will be discussed later. The first version is as follows:

Algorithm F1

1. Set iteration index $j = 0$, and make a set of initial guesses $\{\mathbf{x}_i^{(0)}\}$, $i = 0, 1, \ldots, N$, where $\mathbf{x}_0^{(0)} = \mathbf{x}_0$ (Such a set of initial guesses should not be difficult to make because of the physical meaning of $\mathbf{x}$).

2. Replace $\{\mathbf{x}_i\}$ in the right hand side of Eq. (5.7) by $\{\mathbf{x}_i^{(j)}\}$. Let $\{\mathbf{p}_i^{(j)}\}$ represent the nodal values of the costate satisfying Eq. (5.7) with $\{\mathbf{x}_i\}$ replaced by $\{\mathbf{x}_i^{(j)}\}$.

Set $\mathbf{p}_N^{(j)} = \mathbf{p}_{r_1}^{(k)}$. Find the rest of $\{\mathbf{p}_i^{(j)}\}$, in the reverse order of $i = N-1, \ldots, 0$, by the fixed-point iterations

$$(\mathbf{p}_i^{(j)})^{(l+1)} = \boldsymbol{\eta}_i[(\mathbf{p}_i^{(j)})^{(l)}], \quad l = 0, 1, 2, \ldots \tag{5.11}$$

where $\boldsymbol{\eta}_i$ is dependent on $\{\mathbf{x}_i^{(j)}\}$, and the previously converged solution $\mathbf{p}_{i+1}^{(j)}$ (in backward order) is used in Eq. (5.7) to arrive at $\boldsymbol{\eta}_i$. A reasonable initial guess for above iteration would be $(\mathbf{p}_i^{(j)})^{(0)} = \mathbf{p}_{i+1}^{(j)}$.

3. Replace $\{\mathbf{p}_i\}$ in the right hand side of Eq. (5.6) by the just obtained $\{\mathbf{p}_i^{(j)}\}$. Set $j = j + 1$.

4. Find $\{\mathbf{x}_i^{(j)}\}$ by the following fixed-point iterations: $\mathbf{x}_0^{(j)} = \mathbf{x}_0$, and for $i = 1, 2, \ldots, N$,

$$(\mathbf{x}_i^{(j)})^{(l+1)} = \boldsymbol{\zeta}_i[(\mathbf{x}_i^{(j)})^{(l)}], \quad l = 0, 1, 2, \ldots \tag{5.12}$$

where $\boldsymbol{\zeta}_i$ is based on Eq. (5.6), dependent on $\{\mathbf{p}_i^{(j-1)}\}$ (which have been found in Step 2 above as $\{\mathbf{p}_i^{(j)}\}$ before j is incremented.) and the previously converged value $\mathbf{x}_{i-1}^{(j)}$. An initial guess for above iteration could be $(\mathbf{x}_i^{(j)})^{(0)} = \mathbf{x}_{i-1}^{(j)}$,

5. If the difference between the $\{\mathbf{x}_i^{(j)}\}$ just obtained and the previous $\{\mathbf{x}_i^{(j-1)}\}$ satisfies the convergence criterion

$$\sum_{i=1}^{N} \|\mathbf{x}_i^{(j)} - \mathbf{x}_i^{(j-1)}\| \leq \varepsilon_x \tag{5.13}$$

where $\varepsilon_x > 0$ is a specified small constant, $\{\mathbf{x}_i\} = \{\mathbf{x}_i^{(j)}\}$ will be the converged state trajectory, and the algorithm stops; otherwise we re-define $\{\mathbf{x}_i^{(j)}\}$ by

$$\mathbf{x}_i^{(j)} = \kappa \mathbf{x}_i^{(j)} + (1 - \kappa)\mathbf{x}_i^{(j-1)}, \quad i = 1, \ldots, N \tag{5.14}$$

where $\kappa \neq 0$ is a constant, and the $\mathbf{x}_i^{(j)}$s on the right hand side of the above equation are from the profile $\{\mathbf{x}_i^{(j)}\}$ just found in Step 4. Return to Step 2 with the updated $\{\mathbf{x}_i^{(j)}\}$.

The above algorithm consists of one outer-loop iteration on $\{\mathbf{x}_i\}$, and two inner-loop fixed-point iterations (5.11) and (5.12). The outer-loop is in fact also a fixed-point iteration on $\{\mathbf{x}_i\}$, only modified in Eq. (5.14) to achieve convergence. More on this aspect will be discussed in the next section. In the j-th iteration, the costate history $\{\mathbf{p}_i^{(j)}\}$ is first solved based on the current $\{\mathbf{x}_i^{(j)}\}$ in Step 2. Once $\{\mathbf{p}_i^{(j)}\}$ is obtained, an updated state history $\{\mathbf{x}_i^{(j+1)}\}$ is computed in Step 4. If the $\{\mathbf{x}_i^{(j+1)}\}$ has not converged, the process is repeated. The algorithm relies exclusively on fixed-point iterations. No gradient steps are involved, and the computer software implementation will be extremely simple.

While the outer-loop of Algorithm F1 iterates on $\{\mathbf{x}_i\}$, the algorithm may also be set up to iterate on $\{\mathbf{p}_i\}$ instead in the outer-loop. The two inner fixed-point iterations (5.11) and (5.12) are still the same, except that iteration (5.12) will be before iteration (5.11) this time. Consequently, the places of $\{\mathbf{x}\}$ and $\{\mathbf{p}\}$ are switched in Steps 1, 2, and 5. In order not to repeat the entire algorithm, we will simply state the alternate algorithm as

Algorithm F2

Starting from an initially guessed costate profile $\{\mathbf{p}_i^{(0)}\}$, $\mathbf{p}_N^{(0)} = \mathbf{p}_{\tau_k}^{(k)}$, the algorithm iterates to find $\{\mathbf{p}_i\}$ with similar steps as in Algorithm F1, i.e., with an outer-loop fixed-point iteration and two inner-loop fixed iterations (5.11) and (5.12). The order of the steps in the algorithm will of course be modified accordingly. Once $\{\mathbf{p}_i\}$ is found, the corresponding state will be obtained by Eq. (5.12).

5.2.3 Convergence Analysis for Fixed-Point Iteration (FPI) Method

The critical questions that remain to be answered for the Algorithm F1 (and F2) in the preceding section is about the convergence of the fixed-point iterations. Specifically, will the two inner-loop fixed-point iterations in Eqs. (5.11) and (5.12) converge?

Whether is it possible to achieve the convergence condition (5.13) in the outer loop?

Let us address first the convergence of (5.11) and (5.12). From Eq. (5.6) it is clear that

$$\frac{\partial \boldsymbol{\zeta}_i}{\partial \mathbf{x}_i} = \frac{h}{2} \left. \frac{\partial \mathbf{f}_1(\tau_{i-1/2}, \mathbf{x}, \mathbf{p})}{\partial \mathbf{x}} \right|_{\substack{\mathbf{x} = (\mathbf{x}_i + \mathbf{x}_{i-1})/2 \\ \mathbf{p} = (\mathbf{p}_i + \mathbf{p}_{i-1})/2}} \tag{5.15}$$

Assume that $\|\partial \mathbf{f}_1 / \partial \mathbf{x}\|$ is bounded for all $\mathbf{x} \in \boldsymbol{X}$ and $\mathbf{p} \in \boldsymbol{P}$, where for matrices $\|\cdot\|$ stands for an induced-norm (i.e., by any of the $\mathcal{L}_p$-norms, $p = 1, 2$, or ∞), and $\boldsymbol{X}$ and $\boldsymbol{P}$ are some compact sets in R^6. Furthermore, assume that for $\mathbf{x}_0 \in \boldsymbol{X}$ and any $\mathbf{p}_{\tau_1}^{(k)} \in \boldsymbol{P}_{\tau_1} \subset \boldsymbol{P}$, the solution to the TPBVP in Eqs. (5.2–5.4) satisfies the condition that $\{\mathbf{x}(\tau),\ \mathbf{p}(\tau)\} \in \boldsymbol{X} \cup \boldsymbol{P}$ for any $\tau \in [0,\ \tau_1]$. Therefore on the basis of the continuity of $\mathbf{f}_1$ in Eq. (5.6) and the expression in Eq. (5.15), one can always find an $h_{xmax} > 0$ such that for all $0 < h < h_{xmax}$, $\boldsymbol{\zeta}_i(\cdot)$ maps $\mathbf{x} \to \mathbf{x}$, and $\|\partial \boldsymbol{\zeta}_i / \partial \mathbf{x}_i\| < 1$ along the trajectory of the problem Eqs. (5.2–5.4). By the contraction mapping theory,[44] there exists a unique solution to the equation $\mathbf{x}_i = \boldsymbol{\zeta}_i(\mathbf{x}_i)$ with any $h \in (0,\ h_{xmax})$, and the fixed-point iteration (5.12) converges in $\mathbf{x}$.

Similar observation on the continuity of $\boldsymbol{\eta}_i$ and the assumption of boundedness condition on $\|\partial \mathbf{f}_2 / \partial \mathbf{p}\|$ will lead to the conclusion that there exists an $h_{pmax} > 0$ such that for all $0 < h < h_{pmax}$, $\boldsymbol{\eta}_i(\cdot)$ maps $\mathbf{p} \to \mathbf{p}$, and $\|\partial \boldsymbol{\eta}_i / \partial \mathbf{p}_i\| < 1$. Hence the equation $\mathbf{p}_i = \boldsymbol{\eta}_i(\mathbf{p}_i)$ with any $h \in (0,\ h_{pmax})$ has a unique solution, and the fixed-point iteration (5.11) converges in $\mathbf{p}$. Define

$$h_{max} = \min\ \{h_{xmax},\ h_{pmax}\} \tag{5.16}$$

Then for any choice of the integer $N > 1/h_{max}$, and $h = \tau_1 / N$, both fixed-point iterations in Eqs. (5.11) and (5.12) will converge locally.

In summary, under mild boundedness conditions and for a given τ_1, the choice of a sufficiently fine finite-difference grid will guarantee local convergence of the fixed-point

iterations (5.11) and (5.12) in Algorithm F1 and F2.

To discuss the working and convergence of the outer-loop iteration in Algorithm F1, we group all the equations in Eq. (5.6) and all the equations in Eq. (5.7) respectively as

$$\mathbf{F}_1(\hat{\mathbf{x}}, \hat{\mathbf{p}}) = 0 \tag{5.17}$$

$$\mathbf{F}_2(\hat{\mathbf{x}}, \hat{\mathbf{p}}) = 0 \tag{5.18}$$

where $\hat{\mathbf{x}} = col(\mathbf{x}_1 \dots \mathbf{x}_N) \in R^{6N}$ and $\hat{\mathbf{p}} = col(\mathbf{p}_0 \dots \mathbf{p}_{N-1}) \in R^{6N}$ are the unknowns. With the standard assumption of continuous differentiability on the system dynamics $\mathbf{f}_1$ and $\mathbf{f}_2$ in Eqs. (5.6) and (5.7), all the partial derivatives in $\partial \mathbf{F}_i / \partial \hat{\mathbf{x}}$ and $\partial \mathbf{F}_i / \partial \hat{\mathbf{p}}$, $i = 1, 2$, are continuous. Furthermore, it can be shown that based on the specific forms of Eqs. (5.6) and (5.7), the Jacobians $\partial \mathbf{F}_1 / \partial \hat{\mathbf{x}}$ and $\partial \mathbf{F}_2 / \partial \hat{\mathbf{p}}$ will always be nonsingular for sufficiently small h. Let $\{\hat{\mathbf{x}}^*, \hat{\mathbf{p}}^*\}$ be a pair that satisfies Eqs. (5.17) and (5.18). By the well-known Implicit Function Theorem and Eq. (5.17), there exists a unique function $\hat{\mathbf{x}} = \boldsymbol{\xi}(\hat{\mathbf{p}})$ in a neighborhood of $\{\hat{\mathbf{x}}^*, \hat{\mathbf{p}}^*\}$ such that $\mathbf{F}_1(\boldsymbol{\xi}(\hat{\mathbf{p}}), \hat{\mathbf{p}}) = 0$. Moreover,

$$\frac{\partial \boldsymbol{\xi}(\hat{\mathbf{p}})}{\partial \hat{\mathbf{p}}} = -\left(\frac{\partial \mathbf{F}_1}{\partial \hat{\mathbf{x}}}\right)^{-1} \left(\frac{\partial \mathbf{F}_1}{\partial \hat{\mathbf{p}}}\right) \tag{5.19}$$

From Eq. (5.18) another function $\hat{\mathbf{p}} = \boldsymbol{\theta}(\hat{\mathbf{x}})$ exists also in a neighborhood of $\{\hat{\mathbf{x}}^*, \hat{\mathbf{p}}^*\}$ such that $\mathbf{F}_2(\hat{\mathbf{x}}, \boldsymbol{\theta}(\hat{\mathbf{x}})) = 0$, and

$$\frac{\partial \boldsymbol{\theta}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} = -\left(\frac{\partial \mathbf{F}_2}{\partial \hat{\mathbf{p}}}\right)^{-1} \left(\frac{\partial \mathbf{F}_2}{\partial \hat{\mathbf{x}}}\right) \tag{5.20}$$

Using the above implicit functions, it is not difficult for one to see that Steps 2, 3 and 4 of Algorithm F1 form the relationship

$$\hat{\mathbf{x}}^{(j+1)} = \boldsymbol{\xi}[\hat{\mathbf{p}}^{(j)}] = \boldsymbol{\xi}[\boldsymbol{\theta}(\hat{\mathbf{x}}^{(j)})] \tag{5.21}$$

By Eqs. (5.19) and (5.20)

$$\frac{\partial \boldsymbol{\xi}[\boldsymbol{\theta}(\hat{\mathbf{x}})]}{\partial \hat{\mathbf{x}}} = \left(\frac{\partial \mathbf{F}_1}{\partial \hat{\mathbf{x}}}\right)^{-1} \left(\frac{\partial \mathbf{F}_1}{\partial \hat{\mathbf{p}}}\right) \left(\frac{\partial \mathbf{F}_2}{\partial \hat{\mathbf{p}}}\right)^{-1} \left(\frac{\partial \mathbf{F}_2}{\partial \hat{\mathbf{x}}}\right) := \mathbf{D}_x \tag{5.22}$$

For the fixed-point iteration in (5.21), the condition $\|\mathbf{D}_x\| < 1$ is needed to ensure convergence. But this condition may not be satisfied in some cases. The operation in Eq. (5.14) of Step 5 of Algorithm F1 amounts to the following modification to the fixed-point iteration (5.21)

$$\hat{\mathbf{x}}^{(j+1)} = \kappa\boldsymbol{\xi}[\boldsymbol{\theta}(\hat{\mathbf{x}}^{(j)})] + (1-\kappa)\hat{\mathbf{x}}^{(j)} \tag{5.23}$$

As in the discussion on iteration Eq. (5.1) in Section III, the fixed point of the above equation for any $\kappa \neq 0$ is also the fixed point of Eq. (5.21). The difference is that it is possible to make the fixed-point iteration in Eq. (5.23) converge with some κ even if the one in Eq. (5.21) does not converge.[45] This idea is best illustrated by a scalar case. Consider the problem of finding the solution to the nonlinear scalar equation

$$x = f(x) \tag{5.24}$$

Assume that a solution to above equation exists and $|df/dx| \neq 1$ anywhere. Reformulate this equation as

$$x = \varepsilon f(x) + (1-\varepsilon)x := F(x) \tag{5.25}$$

where $\varepsilon \neq 0$. Evidently the solutions to Eq. (5.24) and Eq. (5.25) are the the same. The gradient of the right hand side of Eq. (5.25) is

$$\frac{dF}{dx} = \varepsilon\frac{df}{dx} + (1-\varepsilon) \tag{5.26}$$

It can be readily verified that $|dF/dx| < 1$ if ε is chosen as follows:

1. If $df/dx > 1$, choose
$$-\frac{2}{df/dx - 1} < \varepsilon < 0$$
2. If $df/dx < 1$, choose
$$0 < \varepsilon < \frac{2}{1 - df/dx}$$

In other words, the mapping on the right hand side of Eq. (5.25) can always be made contractive with an appropriate $\varepsilon \neq 0$, regardless whether or not $f(x)$ is contractive. Therefore the solution to Eq. (5.25), as well as the solution to the original Eq. (5.24), can always be found by the fixed-point iteration

$$x^{(i+1)} = F(x^{(i)}) = \varepsilon f(x^{(i)}) + (1-\varepsilon)x^{(i)},\ i = 0, 1, 2, \ldots,\ \forall x^{(0)} \in R \tag{5.27}$$

The above fixed-point iteration is just a scalar version of Eq. (5.23). In multi-variate case as in Eq. (5.23), of course, the choice of κ will not be as simple. But we are working toward developing similar criteria that are dependent on the Jacobian of $\boldsymbol{\xi}[\boldsymbol{\theta}(\mathbf{x})]$.

Following a parallel discussion leading to Eq. (5.23), we can show that the outer loop in Algorithm F2 is equivalent to a fixed-point iteration on

$$\hat{\mathbf{p}}^{(j+1)} = \kappa\boldsymbol{\theta}[\boldsymbol{\xi}(\hat{\mathbf{p}}^{(j)})] + (1-\kappa)\hat{\mathbf{p}}^{(j)} \tag{5.28}$$

When $\kappa = 1$, the following condition is required for the outer-loop fixed-point iteration to converge

$$\left\| \left(\frac{\partial \mathbf{F}_2}{\partial \hat{\mathbf{p}}}\right)^{-1} \left(\frac{\partial \mathbf{F}_2}{\partial \hat{\mathbf{x}}}\right) \left(\frac{\partial \mathbf{F}_1}{\partial \hat{\mathbf{x}}}\right)^{-1} \left(\frac{\partial \mathbf{F}_1}{\partial \hat{\mathbf{p}}}\right) \right\| := \|\mathbf{D}_p\| < 1 \tag{5.29}$$

Since in general $\|\mathbf{D}_p\| \neq \|\mathbf{D}_x\|$ even if the individual matrices in $\mathbf{D}_x$ and $\mathbf{D}_p$ are the same, there may be cases where for certain vehicles the condition (5.29) is met when Eq. (5.21) is not. In these cases Algorithm F2 will be more suitable.

5.2.4 Approaches Developed for Facilitating the Convergence of Fixed-Point Iteration (FPI)

There are totally three fixed-point iterations in the presented algorithm. In Ref. [9], several approaches have been presented to facilitate the convergence of Fixed-Point Iteration (FPI). Detailed discussion will be given in this part. To avoid possible confusion and for convenience of expression, since then a general fixed-point iteration expression will be used in discussion.

Suppose there is a fixed-point iteration

$$\mathbf{z} = \mathbf{f}(\mathbf{z}) \tag{5.30}$$

Above equation can be expressed as following iteration:

$$\hat{\mathbf{z}}^{j+1} = \mathbf{f}(\mathbf{z}^j), \quad j = 0, 1, 2, \ldots \tag{5.31}$$

where

$$\mathbf{z}^j = \hat{\mathbf{z}}^j, \quad j = 1, 2, \ldots \tag{5.32}$$

This iteration may converge, but we cannot guarantee convergence of the iteration. For a fixed-point iteration, it is possible to facilitate the convergence of the iteration through redefining the fixed-point iteration with an appropriate coefficient θ [45]. With a proper θ, $\mathbf{z}^{j+1}$ can be redefined as

$$\mathbf{z}^{j+1} = \theta\hat{\mathbf{z}}^{j+1} + (1-\theta)\mathbf{z}^j, \quad j = 0, 1, 2, \ldots \tag{5.33}$$

With Eq. (5.31) and Eq. (5.33), we have a new iteration equation:

$$\mathbf{z} = \mathbf{F}(\mathbf{z}) := \theta\mathbf{f}(\mathbf{z}) + (1-\theta)\mathbf{z} \tag{5.34}$$

Evidently the solutions to Eq. (5.34) and Eq. (5.30) are the same. The point is that in the cases where fixed-point iteration (5.30) doesn't converge, we can try to find a proper θ to facilitate the convergence of the fixed-point iteration (5.34). In other words, we are trying to formulate a contraction mapping $\mathbf{z} = \mathbf{F}(\mathbf{z})$ instead of solving the fixed-point iteration (5.30) directly. It should be mentioned that θ can be either a scalar or a diagonal matrix. In the case of diagonal matrix, we denote it as Θ. A detailed discussion on how to choose the proper θ (or Θ) will be given later.

Approach 1: Direct Proportion Method

From Contraction Theorem (Banac Fixed-Point Theorem) [46], if the iteration (5.34) converges to the fixed point, then the iteration sequence ($\mathbf{z}^j$) is a Cauchy sequence, i.e.

$$\|\mathbf{z}^{j+1} - \mathbf{z}^j\| \leq c\|\mathbf{z}^j - \mathbf{z}^{j-1}\| \tag{5.35}$$

where $0 < c < 1$. Substitute Eq. (5.33) and Eq. (5.32) into Eq. (5.35), finally

$$| \theta | \leq \frac{c\|\mathbf{z}^j - \mathbf{z}^{j-1}\|}{\|\hat{\mathbf{z}}^{j+1} - \mathbf{z}^j\|} \tag{5.36}$$

After first two iterations, with Eq. (5.36) we can find a proper θ to make the iteration to converge. However in the case where obtained θ is a small quantity, even $\|\mathbf{z}^j - \mathbf{z}^{j-1}\| \leq \epsilon$ (ϵ is a predetermined small constant), it is possible that $\|\hat{\mathbf{z}}^{j+1} - \mathbf{z}^j\| \gg \epsilon$. Hence, the converged result of $\mathbf{z} = \mathbf{F}(\mathbf{z})$ maybe not the solution of $\mathbf{z} = \mathbf{f}(\mathbf{z})$ with satisfied precision.

Approach 2: Spectral Radius Method

If $\mathbf{F}(\mathbf{z})$ in Eq. (5.34) is a contraction mapping, then by Ref. [45]

$$\|\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}\| < 1. \tag{5.37}$$

From Eq. (5.34)

$$\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}} = \theta \frac{\partial \mathbf{f}}{\partial \mathbf{z}} + (1-\theta)I \tag{5.38}$$

where I is n by n identity matrix. The spectral radius of $\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}$ is:

$$\rho(\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}) := \max_{1 \leq i \leq n} | \lambda_i | \tag{5.39}$$

where λ_i is the $i-th$ eigenvalue of $\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}$. It is known that [45]

$$\rho(\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}) \leq \|\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}\| \tag{5.40}$$

So we can set

$$\rho(\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}) < 1. \tag{5.41}$$

The eigenvalue of $\frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}$ is given by:

$$\det\left[\lambda I - \frac{\partial \mathbf{F}(\mathbf{z})}{\partial \mathbf{z}}\right] = 0 \tag{5.42}$$

From Eq. (5.34) and Eq. (5.42), we have

$$\det\left[\frac{\lambda - 1 + \theta}{\theta} I - \frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}\right] = 0 \tag{5.43}$$

Let $\bar{\lambda}$ be the eigenvalue of $\frac{\partial \mathbf{f}(\mathbf{z})}{\partial \mathbf{z}}$, so

$$\lambda_k = 1 + \theta(\bar{\lambda_k} - 1), \quad k = 1, \ldots, n \tag{5.44}$$

From Eq. (5.41), it is obvious that

$$\mid 1 + \theta(\bar{\lambda_k} - 1) \mid \; < \; 1., \quad k = 1, \ldots, n \tag{5.45}$$

Let

$$a_k \triangleq \bar{\lambda}_k - 1 = b_k + jc_k \tag{5.46}$$

We consider a general case where a_k is a complexnumber, therefore

$$\mid 1 + \theta(b_k + j\ c_k) \mid \; < \; 1., \quad k = 1, \ldots, n \tag{5.47}$$

where $j = \sqrt{-1}$.

- If $b_k = 0$

$$\mid 1 + \theta j\ c_k \mid \; < \; 1. \tag{5.48}$$

  There is no θ which can satisfy above condition.

- If $b_k \; > \; 0$

$$\frac{-2b_k}{{b_k}^2 + {c_k}^2} \; < \; \theta \; < \; 0 \tag{5.49}$$

- If $b_k \; < \; 0$

$$0 \; < \theta \; < \; \frac{-2b_k}{{b_k}^2 + {c_k}^2} \tag{5.50}$$

Using above relationships to calculate through each $a_k(\quad k \; = \; 1, \ldots, n)$, finally a proper range of θ can be concluded. However it is possible that there is no proper scalar θ existing in some cases. For example, no proper scalar θ is available when there is a $b_k = 0$. Or in the case of $b_1 > 0$ but $b_2 < 0$, evidently there is no proper range of θ existing.

Approach 3: Matrix Norm Method

In both of Approach 1 and Approach 2, θ is treated as a scalar. However in our problem formulation, the functions $\mathbf{f}(\mathbf{z})$, $\mathbf{F}(\mathbf{z})$ and the variable $\mathbf{z}$ are all treated as vectors in a general case. Herein it would be a reasonable guess that a diagonal matrix Θ should be used for re-defining. The diagonal matrix Θ can be defined as:

$$\Theta = \begin{pmatrix} \theta_1 & 0 & \dots & 0 \\ 0 & \theta_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & \theta_n \end{pmatrix} \tag{5.51}$$

Rewrite Eq. (5.34) into:

$$\mathbf{z}^{i+1} = \mathbf{F}(\mathbf{z}^i) := \Theta\mathbf{f}(\mathbf{z}^i) + (1 - \Theta)\mathbf{z}^i \tag{5.52}$$

Let $\mathbf{e}^i = \mathbf{z}^i - \mathbf{w}$, $\mathbf{w}$ is the final converged solution of the iteration. It turns out that

$$\mathbf{e}^{i+1} = \Theta\left[\mathbf{f}(\mathbf{z}^i) - \mathbf{f}(\mathbf{w})\right] + (I - \Theta)\mathbf{e}^i \tag{5.53}$$

By the first-order Taylor series expansion, we have

$$\mathbf{e}^{i+1} = (I - \Theta + \Theta G^i)\mathbf{e}^i \tag{5.54}$$

where

$$G^i \triangleq \frac{\partial \mathbf{f}}{\partial \mathbf{z}} \mid_{\mathbf{z}=\xi^i} \tag{5.55}$$

$$\xi_k^i = \mathbf{z}^i{}_k - \phi_k(\mathbf{z}^i{}_k - \mathbf{w}_k), \quad k = 1, \dots, n \tag{5.56}$$

where $0 < \phi_k < 1$. Let

$$M^i \triangleq I - \Theta + \Theta G^i \tag{5.57}$$

From Eq. (5.54), it is evident that the fixed-point iteration $\mathbf{z} = \mathbf{F}(\mathbf{z})$ will monotonically converge if and only if

$$\|M^i\| < 1 \tag{5.58}$$

Let

$$G^i \triangleq \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{pmatrix} \tag{5.59}$$

Therefore

$$I - \Theta + \Theta G^i = \begin{pmatrix} 1-\theta_1+\theta_1 g_{11} & \theta_1 g_{12} & \cdots & \theta_1 g_{1n} \\ \theta_2 g_{21} & 1-\theta_2+\theta_2 g_{22} & \cdots & \theta_2 g_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \theta_n g_{n1} & \theta_n g_{n2} & \cdots & 1-\theta_n+\theta_n g_{nn} \end{pmatrix} \tag{5.60}$$

The 1-norm of the $n \times n$ matrix A is:

$$\|A\| \triangleq \max_k \sum_{j=1}^{n} \mid a_{kj} \mid \tag{5.61}$$

From Eq. (5.58), Eq. (5.60) and above definition of 1-norm, we have

$$\mid 1-\theta_k(1-g_{kk}) \mid + \mid \theta_k \mid \sum_{j \neq k, j=1}^{n} \mid g_{kj} \mid < 1, \quad k = 1, \ldots, n \tag{5.62}$$

If we take the following assumption

$$\mid 1 - g_{kk} \mid \; > \; \sum_{j \neq k, j=1}^{n} \mid g_{kj} \mid, \quad k = 1, \ldots, n \tag{5.63}$$

then it can be shown that [45]

- If $1 - g_{kk} > \sum_{j \neq k, k=1}^{n} \mid g_{kj} \mid$

$$0 < \theta_k < \frac{2}{1 - g_{kk} + \sum_{j \neq k, j=1}^{n} \mid g_{kj} \mid}, \quad k = 1, \ldots, n \tag{5.64}$$

- If $1 - g_{kk} < -\sum_{j \neq k, k=1}^{n} | g_{kj} |$

$$\frac{2}{1 - g_{kk} - \sum_{j \neq k, j=1}^{n} | g_{kj} |} < \theta_k < 0, \quad k = 1, \ldots, n \tag{5.65}$$

Although the conditions Eq. 5.63 seem severe and perhaps unusual they are frequently satisfied in practice. In fact many difference methods for solving nonlinear boundary value problems in ordinary and partial differential equations result in such systems. [45]

We can choose a proper value for each main entry of the diagonal matrix Θ with (5.64) and (5.65). Through redefining the fixed-point iteration (5.30) with this diagonal matrix Θ, it is feasible to get a contraction mapping which will monotonically converge.

Instead of solving the original fixed-point iteration problem directly, we have presented three approaches to redefine the original fixed-point iteration and herein formulate a contraction mapping. The advantages and disadvantages of each approach will be briefly discussed later.

5.2.5 Further Discussion

For the Fixed-Point Iteration (FPI) algorithm presented in Section 5.2.2, there are three fixed-point iterations which include two inner-loop iterations and one outer-loop iteration. To facilitate or even accelerate the convergence of the iteration, all the three iterations can be redefined with the approaches developed in Section 5.2.4. Theoretically, any of those three kind of approaches can be applied to each fixed-point iteration. However during implementation of the algorithm, the computation efficiency and simplicity of the code programming require further evaluation for each approach before making decision on choosing which kind of approach for each fixed-point iteration. A brief evaluation of the three approaches will be given in the following part.

- *For Approach 1: Direct Proportion Method*

  - *Advantages:*

  - There is only a scalar θ required for this approach.
  - It is simple for code programming.
  - It is less computation consuming for each redefining calculation.

- *Disadvantages:*
  - It can not be applied in first two steps of the iteration process.
  - If the calculated θ is too small, the final converged result of the redefined iteration may not the exact solution of the original fixed-point iteration.

- *For Approach 2: Spectral Radius Method*
  - *Advantages:*
    - There is only a scalar θ required for this approach.
    - It can be applied to any steps of the iteration.
  - *Disadvantages:*
    - It is necessary to evaluate the Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{z}}$ and find its eigenvalues at each step of the iteration. Therefore it is more computation consuming.
    - The code programming is more complicated in comparison to Approach 1 and Approach 3.
    - It is possible that there is no appropriate θ existing in some cases.

- *For Approach 3: Matrix Norm Method*
  - *Advantages:*
    - A proper diagonal matrix Θ is always available.
    - It is required to evaluate the Jacobian matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{z}}$, but no eigenvalue solutions are needed.
  - *Disadvantages:*

* The code programming is a little complicated in comparison to Approach 1, but simpler than Approach 2.

From the detailed description in Section 5.2.2, the outer-loop fixed-point iteration is much more complicated in comparison to the two inner-loop fixed-point iterations. After considering carefully about the simplicity of code programming, computation efficiency and precision of the solution, Approach 1 is chosen for redefining the outer-loop iteration and Approach 3 is chosen for redefining the two inner-loop iterations.

5.3 Numerical Approach for Exo-atmospheric Ascent Portion

5.3.1 Analytical Vacuum Optimal Ascent Solutions

The analytical solution to the costate equation (4.7) and approximate analytical solution to state equations Eq. (4.3), summarized in Refs. [18] and [7], are given below for the convenience of the reader. Suppose that the vacuum ascent (second stage and on) starts at τ_1. For $\tau \geq \tau_1$ the costate equation Eq. (4.7) has closed-form solution

$$\begin{aligned}\begin{bmatrix} \mathbf{p}_V(\tau) \\ -\mathbf{p}_r(\tau) \end{bmatrix} &= \begin{bmatrix} \cos[\omega(\tau-\tau_1)]I_3 & \sin[\omega(\tau-\tau_1)]I_3 \\ -\sin[\omega(\tau-\tau_1)]I_3 & \cos[\omega(\tau-\tau_1)]I_3 \end{bmatrix} \begin{bmatrix} \mathbf{p}_{V_1} \\ -\mathbf{p}_{r_1} \end{bmatrix} \\ &:= \Phi(\tau-\tau_1) \begin{bmatrix} \mathbf{p}_{V_1} \\ -\mathbf{p}_{r_1} \end{bmatrix}\end{aligned} \tag{5.66}$$

where $\mathbf{p}_{V_1}$ and $\mathbf{p}_{r_1}$ are the (to-be-determined) initial conditions for the costate at τ_1, and I_3 is a 3×3 unit matrix. Define

$$\mathbf{I}_c(\tau,\tau_1) = \int_{\tau_1}^{\tau} \mathbf{1}_{p_V}(\zeta)\cos(\omega\zeta)A_T(\zeta)d\zeta := \int_{\tau_1}^{\tau} \boldsymbol{i}_c(\zeta)d\zeta \tag{5.67}$$

$$\mathbf{I}_s(\tau,\tau_1) = \int_{\tau_1}^{\tau} \mathbf{1}_{p_V}(\zeta)\sin(\omega\zeta)A_T(\zeta)d\zeta := \int_{\tau_1}^{\tau} \boldsymbol{i}_s(\zeta)d\zeta \tag{5.68}$$

where $\mathbf{1}_{p_V} = \mathbf{p}_V/||\mathbf{p}_V||$. Note that thrust acceleration $A_T(\cdot)$ is time-varying, because the mass is changing. It can be easily verified that starting from the condition $(\mathbf{r}_1\ \mathbf{V}_1)$ at τ_1, the state equations Eq. (4.3) have the following solution[39]

$$\begin{bmatrix} \mathbf{r}(\tau) \\ \mathbf{V}(\tau) \end{bmatrix} = \Phi(\tau - \tau_1) \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{V}_1 \end{bmatrix} + \Gamma(\tau) \begin{bmatrix} \mathbf{I}_c(\tau, \tau_1) \\ \mathbf{I}_s(\tau, \tau_1) \end{bmatrix} \tag{5.69}$$

where

$$\Gamma(\tau) = \frac{1}{\omega} \begin{bmatrix} \sin(\omega\tau)I_3 & -\cos(\omega\tau)I_3 \\ \cos(\omega\tau)I_3 & \sin(\omega\tau)I_3 \end{bmatrix} \tag{5.70}$$

The thrust integrals $\mathbf{I}_c$ and $\mathbf{I}_s$ can be evaluated by a quadrature formula. For example, with $\delta = (\tau - \tau_1)/4$, the Milne's rule leads to

$$\begin{aligned} \mathbf{I}_j(\tau, \tau_1) &= \frac{(\tau - \tau_1)}{90} [7\boldsymbol{i}_j(\tau_1) + 32\boldsymbol{i}_j(\tau_1 + \delta) + 12\boldsymbol{i}_j(\tau_1 + 2\delta) \\ &+ 32\boldsymbol{i}_j(\tau_1 + 3\delta) + 7\boldsymbol{i}_j(\tau_1 + 4\delta)], \ j = c, s \end{aligned} \tag{5.71}$$

Our experiences show that with the nondimensionalization described in Section 3.2, the above quadratures are sufficiently accurate for powered flight up to several hundred seconds. Additional segments in time grid may be used for longer powered flight if necessary.

5.3.2 f and g Series Approach

As mentioned before, the linear gravity approximation is adopted in our dynamics equation. This approximation leads to an analytical solution Eq. (5.69) and Eq. (5.66) for propagating the state and costate variables forward. This closed form solution is convenient for programming and efficient in calculating. However there is unavoidable solution error caused by linear gravity approximation. In some cases this error is too dominant to be ignored. For the case of the large thrust burn, the linear gravity approximation causes a very minor difference on final optimization result because the effect of

gravity on vehicle dynamics is much less than the effect from thrust. However in burn-coast-burn trajectory optimization, the vehicle dynamics during coast and the final low thrust burn are both dominated by the effect from gravity. Especially for the coast, gravity is the only force affecting the dynamics behavior of the vehicle. In these cases, linear gravity approximation will introduce unacceptable error in propagating the state variables with Eq. (5.69).

Let us suppose the state and costate vector at the instant of main engine cutoff τ_{meco} and the coast time are known. The trajectory determination during coast is actually a classical Kepler problem of orbital mechanics. Different iterative and direct approaches are available for solving this problem. Considering the efficiency of computation and the simplicity of code programming, f and g series approach [47] is adopted.

$$\begin{cases} \mathbf{r}_{oms} = f \cdot \mathbf{r}_{meco} + g \cdot \mathbf{V}_{meco} \\ \mathbf{V}_{oms} = \dot{f} \cdot \mathbf{r}_{meco} + \dot{g} \cdot \mathbf{V}_{meco} \end{cases} \tag{5.72}$$

where f and g are defined as:

$$\begin{cases} f = \sum_{n=0}^{\infty} \frac{\tau^n}{n!} F_n \\ g = \sum_{n=0}^{\infty} \frac{\tau^n}{n!} G_n \end{cases} \tag{5.73}$$

$\mathbf{r}_{oms}$ and $\mathbf{V}_{oms}$ are respectively the position and velocity vectors of the vehicle at the begining of 2nd burn. $\mathbf{r}_{meco}$ and $\mathbf{V}_{meco}$ are respectively the position and velocity vectors of the vehicle at the begining of coast (main engine cutoff). It is sufficiently accurate to only adopt first 8 terms in Eq.(5.73). The definition of F_n and G_n can be found in [47]. $\dot{f}$ and $\dot{g}$ are respectively the differentiation of f and g with respect to coast duration.

5.3.3 Multiple-Shooting Formulation

For coast flight where $A_T = 0$, it is more accurate to use the f and g functions/series in orbital mechanics to propagate the state than using Eq. (5.69). Thus the state at

the end of the coast is uniquely defined by the state at the beginning of the coast and the coast time. The costate, on the other hand, may continue to be propagated by Eq. (5.66) during the coast.

Henceforth, the optimal vacuum ascent problem from a given initial condition $\{\mathbf{r}_1, \mathbf{V}_1\}$ with a coast arc is reduced to a root-finding problem. In principle there are 8 unknowns: $\mathbf{p}_{r_1} \in R^3$, $\mathbf{p}_{V_1} \in R^3$, τ_{oms} (the time when coast stops and the third stage burn begins), and τ_f. The corresponding 8 conditions are the k-orbital insertion conditions in Eq. (4.11) and $6-k$ independent transversality conditions (See Section 4.2.3), the switching condition Eq. (4.10), the constraint on final Hamiltonian $H(\tau_f) = 0$, which is from Eq. (4.9).

Unlike in the application to a single powered stage, the approach outlined above, essentially a single-shooting strategy, can suffer from sensitivity problem in multi-burn applications with coast. This is especially so for a long coast and/or very low thrust for the last powered stage. For our stated purposes of in-time ascent guidance planning, the reliability and robustness of the algorithm are of foremost importance. To enhance the convergence of the algorithm, two additional nodes are added to the formulation of the numerical problem. One node is placed at the end of the second powered stage (the first in vacuum flight). Let $\tau_{meco} > \tau_1$ denote the instant when the engine of the second powered stage shuts down (to borrow the terminology in Space Shuttle ascent sequence). The other node is placed at τ_{oms} when the coast ends and the third powered stage begins (OMS burn in the case of the Shuttle). Define the state vector $\mathbf{x} = col(\mathbf{r}\ \mathbf{V})$, costate vector $\mathbf{p} = col(\mathbf{p}_r\ \mathbf{p}_V)$, and the combined vector $\mathbf{y} = col(\mathbf{x}\ \mathbf{p})$. In the interval (τ_1, τ_{meco}), the solution of $\mathbf{y}$ is determined by the condition $\mathbf{y}_1$ at τ_1. We will use $\mathbf{y}^-_{meco}$ to signify the value of $\mathbf{y}$ at τ_{meco}. Introduce two additional to-be-determined vectors $\mathbf{y}^+_{meco}$ and $\mathbf{y}^+_{oms}$. The propagation of $\mathbf{y}$ along the coast arc in the interval $(\tau_{meco},\ \tau_{oms})$ is computed by using $\mathbf{y}^+_{meco}$ as the starting condition. In particular we will denote such propagated value of $\mathbf{y}$ at τ_{oms} by $\mathbf{y}^-_{oms}$. The state and costate along the last powered trajectory in

$(\tau_{oms},\ \tau_f)$ are propagated from $\mathbf{y}^+_{oms}$. The continuity of the state and costate at τ_{meco} and τ_{oms} requires the following two conditions to be met

$$\mathbf{y}^-_{meco} = \mathbf{y}^+_{meco} \tag{5.74}$$

$$\mathbf{y}^-_{oms} = \mathbf{y}^+_{oms} \tag{5.75}$$

Figure 5.2 gives an illustration of the multiple-shooting formulation.

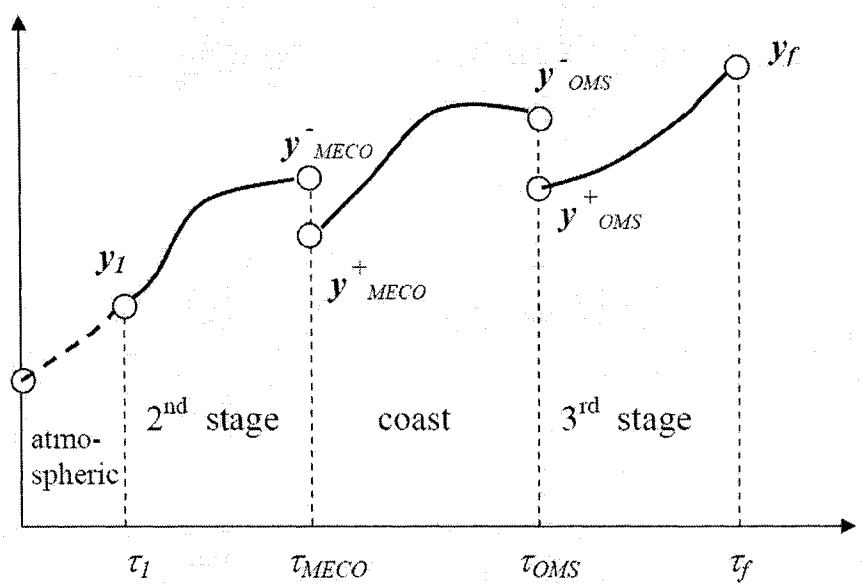

Figure 5.2 Multiple-shooting formulation for optimal vacuum ascent with coast

The above multiple-shooting formulation increases the numbers of unknowns by 24 (the number of scalars included in $\mathbf{y}^+_{meco}$ and $\mathbf{y}^+_{oms}$). The continuity conditions Eqs. (5.74) and (5.75) provide the same number of additional equations. Thus the dimension of the root-finding problem is now 8+24=32.

More explicitly

$$\boldsymbol{S}_1 = \mathbf{p}^-_{meco} - p^+_{meco} \tag{5.76}$$

$$\boldsymbol{S}_2 = \mathbf{x}^-_{meco} - x^+_{meco} \tag{5.77}$$

$$\boldsymbol{S}_3 = \mathbf{p}^-_{oms} - p^+_{oms} \tag{5.78}$$

$$\boldsymbol{S}_4 = \mathbf{x}^-_{oms} - x^+_{oms} \tag{5.79}$$

$$\boldsymbol{S}_5 = \Psi_f \tag{5.80}$$

$$\boldsymbol{S}_6 = H_{NT}(\tau_{oms}) \tag{5.81}$$

where Ψ_f represents the $6-k$ orbital insertion conditions, plus k independent transversality conditions and the constraint on final Hamiltonian $H(\tau_f) = 0$. Let suppose

$$\boldsymbol{S} = [\boldsymbol{S}_1 \;\; \boldsymbol{S}_2 \;\; \boldsymbol{S}_3 \;\; \boldsymbol{S}_4 \;\; \boldsymbol{S}_5 \;\; \boldsymbol{S}_6]^T{}_{32\times 1} = 0 \tag{5.82}$$

where T means transpose. $\boldsymbol{S}$ is the systems of nonlinear equations (32-dimensional). The 32 unknowns variables are: $\mathbf{p}_1$, $\mathbf{p}^+_{meco}$, $\mathbf{x}^+_{meco}$, $\mathbf{p}^+_{oms}$, $\mathbf{x}^+_{oms}$, τ_{oms} and τ_f. Let

$$\mathbf{z} = [\mathbf{p}_1 \;\; {\mathbf{p}^+_{meco}}^T \;\; {\mathbf{x}^+_{meco}}^T \;\; \tau_{oms} \;\; {\mathbf{p}^+_{oms}}^T \;\; {\mathbf{x}^+_{oms}}^T \;\; \tau_f]^T{}_{32\times 1} \tag{5.83}$$

where z is a 32-dimensional variables. Rewrite the Eqs. (5.82) as

$$\boldsymbol{S}(\mathbf{z}) = 0 \tag{5.84}$$

The modified Newton algorithm is well suited to finding the solution of Eqs. (5.84) numerically. Note that the solutions for the state and costate in Section 5.3.1 make the problem completely analytical, from function evaluations to Jacobian computation. The modest dimension of the problem does not constitute a heavy computational requirement that cannot be met more than adequately by a desk top computer. The benefit of this formulation is a more robust algorithm that can more reliably converge even with a relatively long coast and low thrust of the last stage.

5.3.4 Modified Newton Method

For solving the systems of nonlinear equations 5.84, the modified Newton method is one of the best choices. Starting from an initial guess $\mathbf{z}_0$, the search direction $\mathbf{dz}_i$ in the i_{th} iteration is determined by solving the linear algebraic equations

$$\frac{\partial \boldsymbol{S}(\mathbf{z}_{i-1})}{\partial \mathbf{z}} \cdot \mathbf{dz}_i = \boldsymbol{S}(\mathbf{z}_{i-1}),\ i = 0, 1, 2, \ldots \tag{5.85}$$

The step size parameter σ_i is given by:

$$\sigma_i = \max_{0 \leq j} \left\{ \frac{1}{2^j} \;\middle|\; \boldsymbol{S}^T(\mathbf{z}_{i-1} + \frac{\mathbf{dz}_i}{2^j})\boldsymbol{S}(\mathbf{z}_{i-1} + \frac{\mathbf{dz}_i}{2^j}) < \boldsymbol{S}^T(\mathbf{z}_{i-1})\boldsymbol{S}(\mathbf{z}_{i-1}) \right\} \tag{5.86}$$

where $0 < \sigma_i \leq 1$.

Therefore, starting from $\sigma_i = 1$, parameter σ_i is halved repeatedly if necessary until Eq. (5.86) is satisfied. Once σ_i is determined, the update is given by:

$$\mathbf{z}_i = \mathbf{z}_{i-1} + \sigma_i \mathbf{dz}_i \tag{5.87}$$

With Eq. (5.86), the search step size ensures that the sequence $\{\|\boldsymbol{S}(\mathbf{z}_i)\|\}$ is monotonically decreasing. Convergence is achieved when $\|\boldsymbol{S}(\mathbf{z}_i)\|$ is no greater than a pre-selected tolerance.

One more important issue left is how to evaluate the Jacobian matrix $\frac{\partial \boldsymbol{S}(\mathbf{z})}{\partial \mathbf{z}}$. With application of the analytical solutions for the state and costate in Section 5.3.1, Jacobian matrix $\frac{\partial \boldsymbol{S}(\mathbf{z})}{\partial \mathbf{z}}$ can be obtained analytically.

$$\frac{\partial \boldsymbol{S}(\mathbf{z})}{\partial \mathbf{z}} = \begin{pmatrix} \frac{\partial S_1}{\partial \mathbf{p}_1} & -I & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial S_2}{\partial \mathbf{p}_1} & 0 & -I & 0 & 0 & 0 & 0 \\ 0 & \frac{\partial S_3}{\partial \mathbf{p}_{meco}} & 0 & \frac{\partial S_3}{\partial \tau_{oms}} & -I & 0 & 0 \\ 0 & 0 & \frac{\partial S_4}{\partial \mathbf{x}_{meco}} & \frac{\partial S_4}{\partial \tau_{oms}} & 0 & -I & 0 \\ 0 & 0 & 0 & \frac{\partial S_5}{\partial \tau_{oms}} & \frac{\partial S_5}{\partial \mathbf{p}_{oms}} & \frac{\partial S_5}{\partial \mathbf{x}_{oms}} & \frac{\partial S_5}{\partial \tau_f} \\ 0 & 0 & 0 & 0 & \mathbf{x}_{oms} & \mathbf{p}_{oms} & 0 \end{pmatrix} \tag{5.88}$$

where I is a 6×6 identity matrix. The detailed expressions of $\frac{\partial \boldsymbol{S}_1}{\partial \mathbf{p}_1}$, $\frac{\partial \boldsymbol{S}_2}{\partial \mathbf{p}_1}$, $\frac{\partial \boldsymbol{S}_3}{\partial \mathbf{p}_{meco}}$, $\frac{\partial \boldsymbol{S}_3}{\partial \tau_{oms}}$, $\frac{\partial \boldsymbol{S}_4}{\partial \tau_{oms}}$, $\frac{\partial \boldsymbol{S}_5}{\partial \tau_{oms}}$, $\frac{\partial \boldsymbol{S}_5}{\partial \mathbf{p}_{oms}}$, $\frac{\partial \boldsymbol{S}_5}{\partial \mathbf{x}_{oms}}$ and $\frac{\partial \boldsymbol{S}_5}{\partial \tau_f}$ can be found in Appendix A.

CHAPTER 6. VERIFICATION AND VALIDATION

6.1 Launch Vehicle Model

A generic launch vehicle model is used for this research. The launch vehicle is designed to have three powered stages, respectively with the Booster Engines, Main Engines and OMS engines. In the first two powered stages, the vehicle's engines run out of the all carried propellant for each stage. The thrust, mass flow rate and carried propellant for each stage can be found in Table 6.1.

Table 6.1 A generic launch vehicle prototype specifications

Powered Stage	Thrust (N)	Mass Flow Rate (kg/sec)	Carried Propellant (kg)
1st Powered Stage	2303000.0	275.0	43008.0
2nd Powered Stage	1535333.33	183.33	38168.39
3rd Powered Stage	115150.0	13.75	6000.0

The reference area of the launch vehicle is: $S_{ref} = 149.388(m^2)$, the axial and normal aerodynamic coefficients C_A and C_N are given by (See Section 3.2):

$$C_A = C_{A0}(M_a) + C_{A1}(M_a)\alpha + C_{A2}(M_a)\alpha^2 \tag{6.1}$$

$$C_N = C_{N0}(M_a) + C_{N1}(M_a)\alpha + C_{N2}(M_a)\alpha^2 \tag{6.2}$$

where M_a is the Mach number, α is the angle of attack. Calculations of C_{A0}, C_{A1}, C_{A2}, C_{N0}, C_{N1} and C_{N2} can be found in APPENDIX B.

The maximum acceleration limit is $4g$, where g is the standard gravity acceleration. The peak dynamic pressure constraint is 18194.0 N/m^2 (380.0 psf).

All simulation tests are performed on a desk computer with a 1GHz Pentium 4 CPU and 512MB of RAM.

6.2 Verification of the Analytical Multiple-Shooting (AMS) Method

The Analytical Multiple-Shooting (AMS) method developed in Section 5.3.1 is verified by a well-established Sequential Quadratic Programming (SQP) method which is available in MATLAB. The verification is performed by comparing the results obtained from the two methods. Specifically, the model of a generic launch vehicle in Section 6.1 is used. The initial condition is fixed to that of the vehicle when it just clears the dense atmosphere (at the burnout of a fixed atmospheric ascent profile). where the altitude is at 59.0767 km and the inertial velocity at 2,164.76 m/s. The second stage burns out after another fixed period of 208.19 seconds. Then the coast follows immediately. The coast time and the burn time of the third stage are determined as part of the optimal solution. At the beginning of the coast, the vehicle mass is 41,000 kg. The thrust for the second stage is 1,535,333 N, and for the third stage 115,150 N.

The SQP solution is obtained as a parameter optimization problem in which the optimization variables are the 8 unknowns listed at the begining of Section 5.3.3. The cost function of maximization is the final mass $m(\tau_f)$. The constraints are the k orbital insertion conditions. The analytical solutions given in Section 5.3.1 are used to compute the needed quantities, including the primer vector $\mathbf{p}_V$ since it is required to determine the thrust direction. Note that this is still a direct approach to the optimization problem, because $m(\tau_f)$ is directly optimized and no transversality conditions and the switching condition Eq. (4.10) is enforced. In contrast, the AMS is an indirect method based on the necessary conditions in optimal control theory.

From the given initial conditions, the optimal trajectory for the second stage, coast

arc and the third stage is found by using the AMS and SQP method for two kinds of missions.

Mission 1 : 4 orbital insertion conditions applied: elliptic orbit (e= 0.005), inclination i= 51.6 (deg), flight path angle $\gamma = 0$ at insertion point, perigee altitude is varied from 152.383325 (km) in several increments to 550.383325 (km).

Mission 2 : 5 orbital insertion conditions applied: circular orbit, inclination i= 51.6 (deg), flight path angle $\gamma = 0$ at insertion point, logitude of ascending node Ω = -104.0 (deg), altitude is varied from 185.2 (km) in several increments to 485.2 (km).

All the missions are for launch from the Kennedy Space Center (KSC).

Comparison of Results from AMS method and SQP method

Table 6.2 and Table 6.3 summarizes the results in terms of the coast time, the burn time of the third stage, and the final mass in every case.

Table 6.2 Comparison of results from AMS method and SQP method for Mission 1

perigee altitude (km)	method	coast time (sec)	3rd stage burn-time (sec)	m(τ_f) (kg)
152.383325	AMS	161.8566	151.8487	38912.080
	SQP	160.5619	151.8470	38912.104
251.883325	AMS	332.2385	184.8386	38458.468
	SQP	335.4252	184.7820	38459.248
351.383325	AMS	415.2198	216.3833	38024.729
	SQP	428.6322	216.0361	38029.504
450.883325	AMS	495.1613	280.0214	37149.705
	SQP	529.5226	278.1695	37175.170
550.383325	AMS	513.1059	313.0745	36695.225
	SQP	554.7619	310.2392	36734.211

Table 6.3 Comparison of results from AMS method and SQP method for Mission 2

insertion altitude (km)	method	coast time (sec)	3rd stage burn-time (sec)	m(τ_f) (kg)
185.2	AMS	237.335	161.571	38786.471
	SQP	248.272	159.232	38810.560
285.2	AMS	359.936	193.462	38339.895
	SQP	374.945	190.353	38382.644
385.2	AMS	427.992	225.084	37905.101
	SQP	436.261	221.420	37955.475
485.2	AMS	469.771	257.048	37465.595
	SQP	504.741	252.216	37532.030

It can be seen from Table 6.2 and Table 6.3 that the AMS and SQP methods yield very similar results. As expected, as the orbital insertion altitude gets higher, the coast time increases as well as the burn time of the last stage. In the cases reported, the AMS solutions have slightly less final masses. But these small discrepancies could easily vary in different direction and are not the proclaimed advantage of the SQP method. When changes in mission and vehicle propulsion characteristics increase the coast time, the problem will become more sensitive, the SQP method can and will fail to find the solution (we have encountered such cases). This sensitivity, on the other hand, is precisely what the AMS algorithm is designed to overcome.

Comparison of Results from Burn-Burn and Burn-Coast-Burn

The Mission 1 and Mission 2 can be performed with either two powered stage without coast or an optimal coast in between the two powered stages. The final mass m(τ_f) with these two ways for different insertion altitude are summarized in Table 6.4 and Table 6.5. In these two tables, all final mass m(τ_f) for "with coast" are calculated by the Analytical Multiple-Shooting Method. While all final mass m(τ_f) for "without coast"

are calculated by the two stages classic vaccum guidance algorithm. $\Delta m(\tau_f)$ represents the fuel usage difference between "with coast" and "without coast".

Table 6.4 Comparison of final mass with burn-burn and burn-coast-burn for Mission 1

insertion altitude (km)	152.383	251.883	351.383	450.883
m(τ_f) without coast (kg)	38858.987	37971.210	37146.290	36450.290
m(τ_f) with coast (kg)	38912.080	38458.468	38024.729	37149.705
$\Delta m(\tau_f)$(kg)	53.093	487.258	878.439	699.415

Table 6.5 Comparison of final mass with burn-burn and burn-coast-burn for Mission 2

insertion altitude (km)	185.2	285.2	385.2	485.2
m(τ_f) without coast (kg)	38517.495	37668.021	36907.082	36253.854
m(τ_f) with coast (kg)	38786.471	38339.895	37905.101	37465.595
$\Delta m(\tau_f)$(kg)	268.976	368.685	998.019	1211.741

From Table 6.4 and Table 6.5, it is confirmed clearly that it is more efficient on fuel usage to include an optimal coast in between the two powered stages than the way without coast. With the increase of the insertion altitude, the most difference on fuel usage even can be around $1200kg$!

6.3 Verification of the Fixed-Point Iteration (FPI) Algorithm

The fixed-point algorithm is verified by performing the generation of complete ascent trajectories from lift-off to orbital insertion for several different missions. All the missions are for launch from the Kennedy Space Center (KSC). The target orbits are either circular or elliptical at an inclination of 51.6 deg or 28.5 deg. To verify the effectiveness of the presented algorithm, the Finite Difference (FD) method in Ref. [7] and the Fixed-Point Iteration (FPI) algorithm are respectively used to generate the endo-atmospheric ascent trajectory. A path constraint on the peak dynamic pressure

constraint at 18,194 N/m^2 (380 psf) is imposed. The maximum thrust acceleration limit is $4g$. The Analytical Multiple-Shooting algorithm is responsible for the solution of the exo-atmospheric portion of the ascent trajectory. The integration algorithm in Section 5.1 combines the endo-atmospheric and vacuum algorithms to obtain the complete ascent trajectory. Once the trajectory is found, the complete information including the optimal guidance command profile is known. Closed-loop simulations are also carried out in which the same algorithms are used to update the trajectory/commands in 1 HZ cycle (that is, re-solving the remaining ascent trajectory with the current condition as the initial condition). The updated commands are used to simulate the flight of the vehicle in that cycle.

6.3.1 Comparison of the Efficiency of Three Approaches for Facilitating the Convergence of Fixed-Point Iteration

In Section 5.2.4, three different approaches (Direct Proportion Method, Spectral Radius Method and Matrix Norm Method) are developed to facilitate the convergence of the Fixed-Point iteration. The advantages and disadvantages of these three approaches are discussed in Section 5.2.5. A simple test is explored to compare the efficiency of the three different approaches. 100, 60 and 40 nodes are placed respectively along an endo-atmospheric trajectory. The maximum iterations (denoted as K_{max}) , minimum iterations (denoted as K_{min}) and average iterations (denoted as K_{avg}) during first round iteration on co-state $\mathbf{p}$ with different facilitating approaches and the original Fixed-Point Iteration are summarized in Table 6.6, Table 6.7, and Table 6.8.

In Table 6.7, the Fixed-Point iteration on No.55 node fails to converge for Original Fixed-Point Iteration. In Table 6.8, the Fixed-Point iteration on No.39 node fails to converge for Original Fixed-Point Iteration.

Table 6.6 Comparison of efficiency with different FPI convergence facilitating approaches (100 nodes)

Approach	K_{max}	K_{min}	K_{avg}
Direct Proportion Method	147	53	111
Spectral Radius Method	37	5	17
Matrix Norm Method	32	6	19
Original Fixed-Point Iteration	36	4	14

Table 6.7 Comparison of efficiency with different FPI convergence facilitating approaches (60 nodes)

Approach	K_{max}	K_{min}	K_{avg}
Direct Proportion Method	155	73	121
Spectral Radius Method	150	7	35
Matrix Norm Method	51	15	36
Original Fixed-Point Iteration	N/A	N/A	N/A

Table 6.8 Comparison of efficiency with different FPI convergence facilitating approaches (40 nodes)

Approach	K_{max}	K_{min}	K_{avg}
Direct Proportion Method	173	58	116
Spectral Radius Method	97	10	59
Matrix Norm Method	81	18	50
Original Fixed-Point Iteration	N/A	N/A	NA

From the data listed in Table 6.6, Table 6.7 and Table 6.8, we can draw following conclusions:

- It is confirmed that the three approaches are effective on facilitating the convergence of Fixed-Point iteration in the cases where the Original Fixed-Point Iteration fails to converge.

- It is possible that Original Fixed-Point Iteration can converge faster than Re-defined Fixed-Point Iteration. On the other hand, Re-defined Fixed-Point Iteration can converge while Original Fixed-Point Iteration fails to converge for some cases.

Hence it is necessary to develop the approaches to facilitate the convergence of the Fixed-Point Iteration.

- It can be seen that Spectral Radius Method and Matrix Norm Method are much more efficient in comparison to Direct Proportion Method. However as mentioned before, Spectral Radius Method may not work in some cases. (See Section 5.2.5) Therefore for the two inner loop Fixed-Point iterations, Matrix Norm Method is the best option.

- It can be found that reduction of the nodes will most likely introduce more iteration for each inner-loop Fixed-Point iteration. In other words, it reduces the number of nodes for calculating while leads to spend more computation resource on each node.

6.3.2 Comparison of Open-loop Simulations between Fixed-Point Iteration (FPI) and Finite Difference (FD) Algorithms

Mission 3 : 4 orbital insertion conditions applied: circular orbit, inclination i= 51.6 (deg), flight path angle $\gamma = 0$ at insertion point, insertion altitude is varied from 185.2 (km) in several increments to 485.2 (km).

Mission 4 : 4 orbital insertion conditions applied: circular orbit, inclination i= 28.5 (deg), flight path angle $\gamma = 0$ at insertion point, insertion altitude is varied from 185.2 (km) in several increments to 485.2 (km).

Mission 5 : 5 orbital insertion conditions applied: circular orbit, inclination i= 51.6 (deg), flight path angle $\gamma = 0$ at insertion point, longitude of ascending node Ω = -104.0 (deg), insertion altitude is varied from 185.2 (km) in several increments to 485.2 (km).

All the missions are for launch from the Kennedy Space Center (KSC). The open-loop simulations are performed for different missions. FPI and FD algorithms are respectively used to generate the trajectory for endo-atmopsheric portion, while AMS algorithm is

responsible for generating the trajectory of exo-atmosperic portion. The two portions are combined with the integration algorithm presented in Section 5.1. All open-loop simulation results are listed in Table 6.9, Table 6.10 and Table 6.11.

The quantities with r(τ_f) (sec) listed in the tables are the orbital insertion altitudes; τ_1 (sec) is the instant switching from atmospheric flight to vacuum flight; $\tau_{meco} - \tau_1$ (sec) is the duration of 2$^{\text{nd}}$ powered stage; $\tau_f - \tau_{oms}$ (sec) is the duration of 3$^{\text{rd}}$ powered stage; τ_f (sec) is the time to go while m(τ_f) (kg) is the final mass of the vehicle.

It can be seen from Table 6.9 to Table 6.11, that the FPI and FD methods yield almost same results in open-loop simulations for different test cases. FD is a well established numerical method for endo-atmospheric ascent guidance problem. Therefore it has confirmed the effectiveness of FPI algorithm developed in this dissertation in the aspect of open-loop simulations. The further closed-loop simulations comparison between the FD and FPI will be presented later.

Table 6.9 Comparison of open-loop results from FPI algorithm and FD algorithm for Mission 3

r(τ_f) (km)	method	τ_1	$\tau_{meco} - \tau_1$	coast time	$\tau_f - \tau_{oms}$	τ_f	m(τ_f) (kg)
185.2	FD	156.39	202.74	335.88	144.95	839.97	40006.91
	FPI	156.39	202.74	336.02	144.91	840.05	40007.55
285.2	FD	156.39	202.74	425.29	178.93	963.35	39539.75
	FPI	156.39	202.74	425.17	178.99	963.28	39538.89
385.2	FD	156.39	202.74	470.41	212.51	1042.05	39078.03
	FPI	156.39	202.74	470.47	212.50	1042.09	39078.17
485.2	FD	156.39	202.74	493.71	246.33	1099.17	38612.95
	FPI	156.39	202.74	493.73	246.34	1099.19	38612.85

Table 6.10 Comparison of open-loop results from FPI algorithm and FD algorithm for Mission 4

r(τ_f) (km)	method	τ_1	$\tau_{meco} - \tau_1$	coast time	$\tau_f - \tau_{oms}$	τ_f	m(τ_f) (kg)
185.2	FD	156.39	202.74	403.97	122.79	885.89	40311.64
	FPI	156.39	202.74	403.41	122.95	885.49	40309.42
285.2	FD	156.39	202.74	493.19	155.01	1007.33	39868.57
	FPI	156.39	202.74	492.73	155.16	1007.02	39866.62
385.2	FD	156.39	202.74	538.30	186.94	1084.37	39429.54
	FPI	156.39	202.74	537.86	187.10	1084.09	39427.44
485.2	FD	156.39	202.74	561.88	219.50	1140.51	38981.87
	FPI	156.39	202.74	561.49	219.64	1140.26	38979.95

Table 6.11 Comparison of open-loop results from FPI algorithm and FD algorithm for Mission 5

r(τ_f) (km)	method	τ_1	$\tau_{meco} - \tau_1$	coast time	$\tau_f - \tau_{oms}$	τ_f	m(τ_f) (kg)
185.2	FD	156.39	202.74	337.55	148.19	844.87	39962.35
	FPI	156.39	202.74	337.55	148.22	844.90	39962.01
285.2	FD	156.39	202.74	424.80	181.33	965.26	39506.75
	FPI	156.39	202.74	424.63	181.38	965.14	39505.97
385.2	FD	156.39	202.74	469.57	214.37	1043.07	39052.40
	FPI	156.39	202.74	469.52	214.41	1043.06	39051.88
485.2	FD	156.39	202.74	492.85	247.84	1099.83	38592.15
	FPI	156.39	202.74	492.85	247.88	1099.86	38591.61

6.3.3 Orbital Insertion Precision with Fixed-Point Iteration Algorithm

Closed-loop simulations with FPI algorithm are performed for *Mission 6*, *Mission 7* and *Mission 7* to test the orbital insertion precision. More closed-loop simulation results will be presented later to compare with the results from closed-loop simulation with FD algorithm.

Mission 6 : elliptical orbit (eccentricity e= 0.01), inclination i = 51.6 (deg), perigee altitude = 431.22 (km), free Ascending Node.

Mission 7: circular orbit, inclination i = 28.5 (deg), insertion altitude = 285.2 (km), free Ascending Node.

Mission 8 : circular orbit, inclination i= 51.6 (deg), insertion altitude = 285.2 (km), longitude of ascending node Ω = -104.0 (deg).

In Table 6.12, the quantities with Δ are the differences between the final orbital insertion conditions in closed-loop simulations and the required ones (where i for inclination, γ for flight path angle, e for eccentricity, and Ω for longitude of ascending node of the target orbit, respectively).

Table 6.12 Orbital insertion precision from closed-loop simulations with FPI algorithm

mission	Δr_f (m)	ΔV_f (m/s)	$\Delta\gamma$ (deg)	Δi (deg)	Δe	$\Delta\Omega$ (deg)
Mission 6	1.020E-1	-1.133E-1	-5.500E-3	-1.200E-6	3.020E-5	n/a
Mission 7	7.000E-3	-3.848E-4	1.840E-4	3.210E-6	1.020E-7	n/a
Mission 8	2.000E-3	2.532E-2	-3.045E-4	-1.670E-6	8.431E-6	-5.709E-6

6.3.4 Comparison of Closed-loop Simulations between Fixed-Point Iteration (FPI) and Finite Difference (FD) Algorithms

The open-loop simulations just generate the nominal optimal trajectories. For onboard generation of the optimal trajectories, the closed-loop simulation mode is required to test the guidance algorithms. In closed-loop simulation mode, the nominal optimal trajectory is updated once every guidance cycle. At each instant for calculating, the current state information is used as the initial condition in searching new nominal optimal trajectory. Its the reason to call this guidance strategy as closed-loop.

The closed-loop simulations are performed for *Mission 3*, *Mission 4* and *Mission 5*.

The complete ascent trajectories from lift-off to orbital insertion are generated for different missions. FPI and FD algorithms are respectively used to generate the trajectory for endo-atmopsheric portion, while AMS algorithm is responsible for generating the trajectory of exo-atmosperic portion. The two portions are combined with the integration algorithm presented in Section 5.1. All closed-loop simulation results are listed in Table 6.13 to Table 6.15.

The quantities with $\mathrm{r}(\tau_f)$ listed in the tables are the orbital insertion altitudes; τ_1 is the instant switching from atmospheric flight to vacuum flight; $\tau_{meco} - \tau_1$ is the duration of 2^{nd} powered stage; $\tau_f - \tau_{oms}$ is the duration of 3^{rd} powered stage; τ_f is the time to go. All previous terms are in the unit of *sec.*. While $\mathrm{m}(\tau_f)$ is the final mass of the vehicle.

From the closed-loop simulation results presented in Table 6.13 to Table 6.15, it can be seen that FD and FPI methods provide the very similar closed-loop profiles for different test cases. It further confirms the effectiveness of the FPI algorithm in terms of the results of closed-loop simulations.

Table 6.13 Comparison of closed-loop results from FPI algorithm and FD algorithm for Mission 3

$\mathrm{r}(\tau_f)$ (km)	method	τ_1	$\tau_{meco} - \tau_1$	coast time	$\tau_f - \tau_{oms}$	τ_f	$\mathrm{m}(\tau_f)$ (kg)
185.2	FD	171.87	202.74	204.18	206.28	785.07	39163.60
	FPI	172.21	202.74	202.07	207.64	784.65	39145.00
285.2	FD	171.83	202.74	294.51	240.57	909.65	38692.19
	FPI	172.29	202.74	291.28	242.78	909.09	38661.76
385.2	FD	172.61	202.74	338.31	278.58	992.25	38169.48
	FPI	172.35	202.74	341.14	277.11	993.34	38189.75
485.2	FD	171.88	202.74	372.98	309.92	1057.52	37738.53
	FPI	172.43	202.74	372.75	311.05	1058.97	37723.09

Table 6.14 Comparison of closed-loop results from FPI algorithm and FD algorithm for Mission 4

$r(\tau_f)$ (km)	method	τ_1	$\tau_{meco} - \tau_1$	coast time	$\tau_f - \tau_{oms}$	τ_f	$m(\tau_f)$ (kg)
185.2	FD	171.78	202.74	250.58	185.04	810.18	39455.65
	FPI	172.10	202.74	248.24	186.05	809.13	39441.82
285.2	FD	171.85	202.74	345.93	219.07	939.59	38987.78
	FPI	172.26	202.74	342.05	220.87	937.92	38963.08
385.2	FD	171.90	202.74	396.37	252.17	1023.17	38532.66
	FPI	172.28	202.74	394.35	253.89	1023.26	38509.04
485.2	FD	171.98	202.74	427.13	285.39	1087.23	38075.95
	FPI	172.32	202.74	424.00	286.97	1086.04	38054.12

Table 6.15 Comparison of closed-loop results from FPI algorithm and FD algorithm for Mission 5

$r(\tau_f)$ (km)	method	τ_1	$\tau_{meco} - \tau_1$	coast time	$\tau_f - \tau_{oms}$	τ_f	$m(\tau_f)$ (kg)
185.2	FD	171.84	202.74	208.33	208.25	791.15	39136.60
	FPI	172.21	202.74	205.74	209.73	790.41	39116.17
285.2	FD	171.86	202.74	296.08	242.42	920.22	38666.68
	FPI	172.24	202.74	301.50	244.05	920.53	38644.27
385.2	FD	171.91	202.74	345.77	276.22	996.63	38202.04
	FPI	172.26	202.74	345.26	277.70	997.96	38181.67
485.2	FD	171.94	202.74	376.73	310.05	1061.46	37736.83
	FPI	172.37	202.74	372.94	311.80	1059.84	37712.82

Figure 6.1 to 6.18 give the closed-loop profiles respectively from FD and FPI algorithms for following three cases:

- Case1: Mission 3, insertion altitude is 485.2 (km)
- Case2: Mission 5, insertion altitude is 485.2 (km)
- Case3: Mission 4, insertion altitude is 485.2 (km)

Figure 6.1, 6.7 and 6.13 show the three dimensional ascent trajectories and ground tracks for Case1, Case2 and Case3 respectively.

Figure 6.2, 6.8 and 6.14 show closed-loop profiles of the altitude and inertial velocity for Case1, Case2 and Case3 respectively.

Figure 6.3, 6.9 and 6.15 give the closed-loop profiles of the flight path angle γ and the angle of attack α for Case1, Case2 and Case3 respectively. The angle of attack is a key parameter for atmospheric portion of the ascent.

Figure 6.4, 6.10 and 6.16 present the closed-loop profiles of the pitch angle θ and yaw angle ψ of the vehicle. The vehicle body attitude is determined by the pitch and yaw angles which in this case are with respect to an inertial launch frame.

Figure 6.5, 6.11 and 6.17 give the variations of the engine throttle and axial thrust acceleration in closed-loop simulations for Case1, Case2 and Case3 respectively. Note that the dynamic pressure constraint is enforced closely. The "throttle back" can be seen in the engine throttle profile in the first stage (starting at about $t = 20$ sec) which is due to the closed-loop throttle adjustment to enforce the dynamic pressure constraint.

Figure 6.6, 6.12 and 6.18 give the variations of the dynamic pressure q and the product of the sideslip angle and dynamic pressure $q\beta$ in closed-loop simulations. The zero-sideslip formulation is used in this research, therefore the $q\beta$ should be zero for the converged optimal solutions.

From the results of closed-loop simulations, it can be seen that the Fixed-Point Iteration algorithm and Finite Difference algorithms present very similar simulation results. The FD method is an existing algorithm for the closed-loop endo-atmospheric ascent guidance. Through comparing the open-loop and closed-loop simulations with the FPI algorithm and those with the FD algorithm, it is verified that FPI algorithm is an effective new approach for the closed-loop endo-atmospheric ascent guidance.

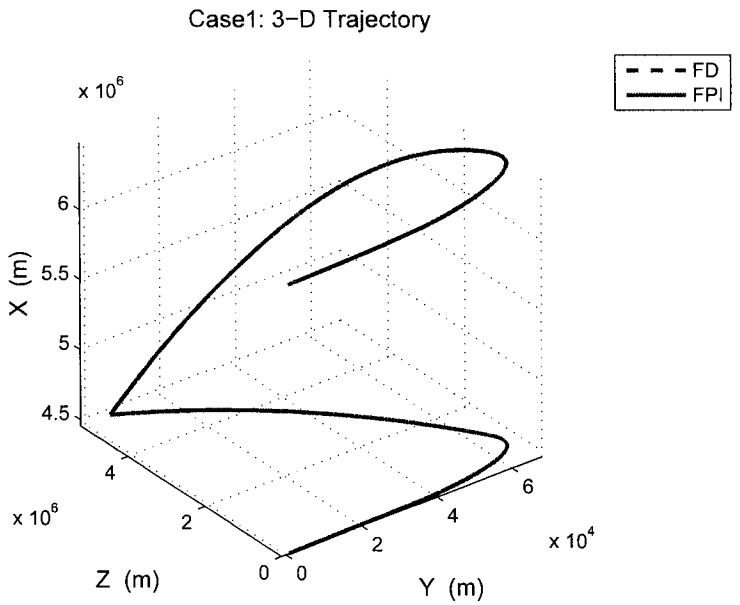

Figure 6.1 Complete 3-D ascent trajectory and ground track for Case1

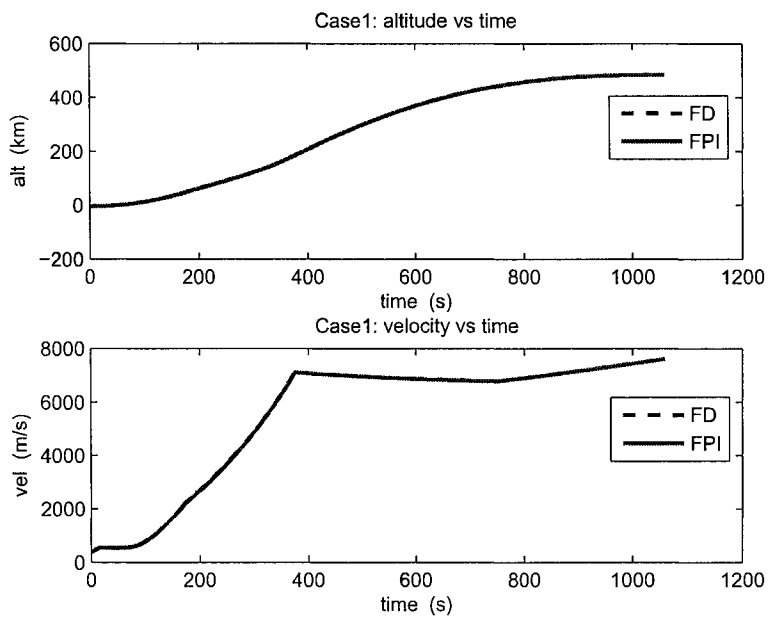

Figure 6.2 Altitude and velocity profiles for Case1

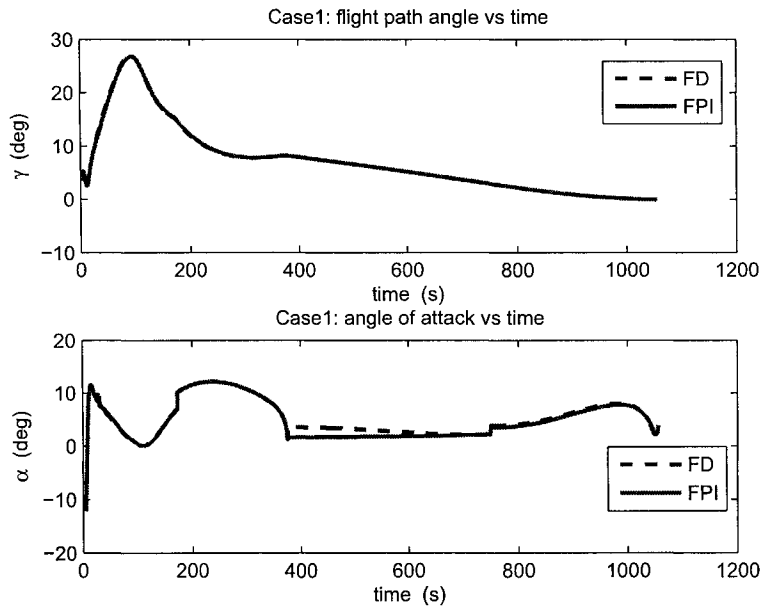

Figure 6.3 Flight path angle and angle of attack profiles for Case1

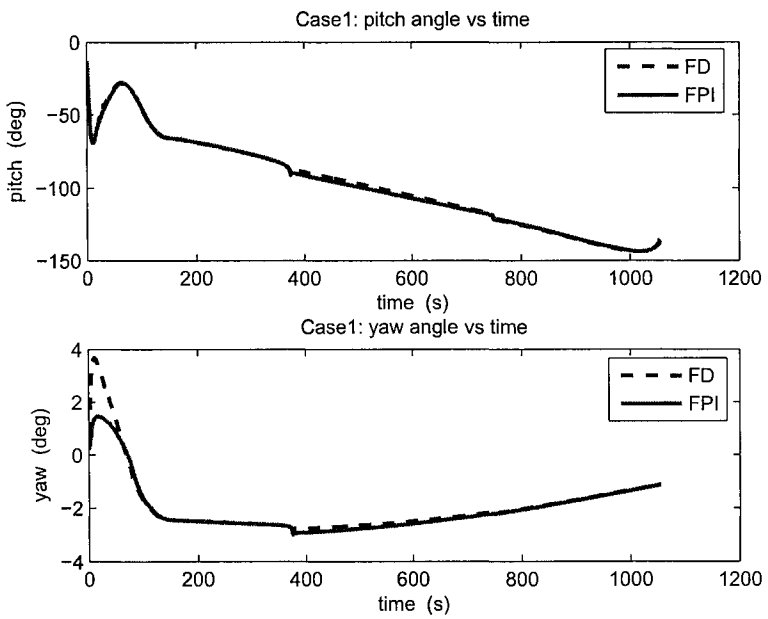

Figure 6.4 Pitch angle and yaw angle profiles for Case1

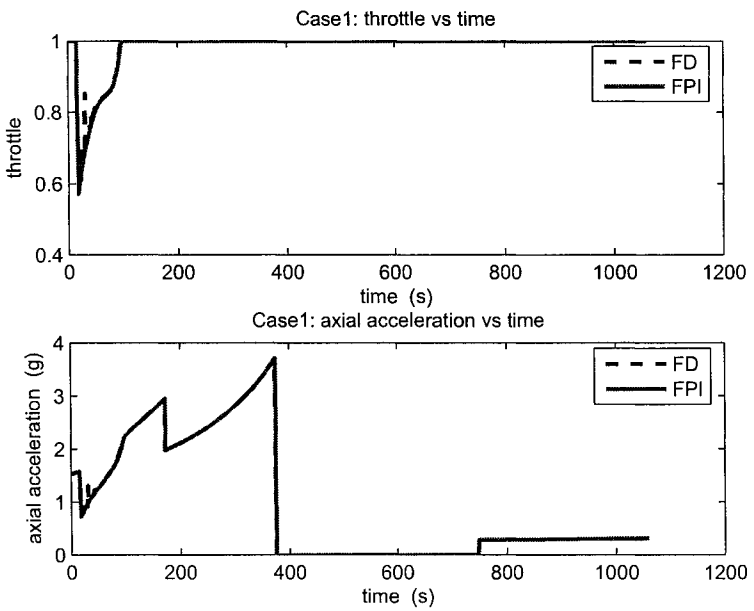

Figure 6.5 Throttle and axial acceleration profiles for Case1

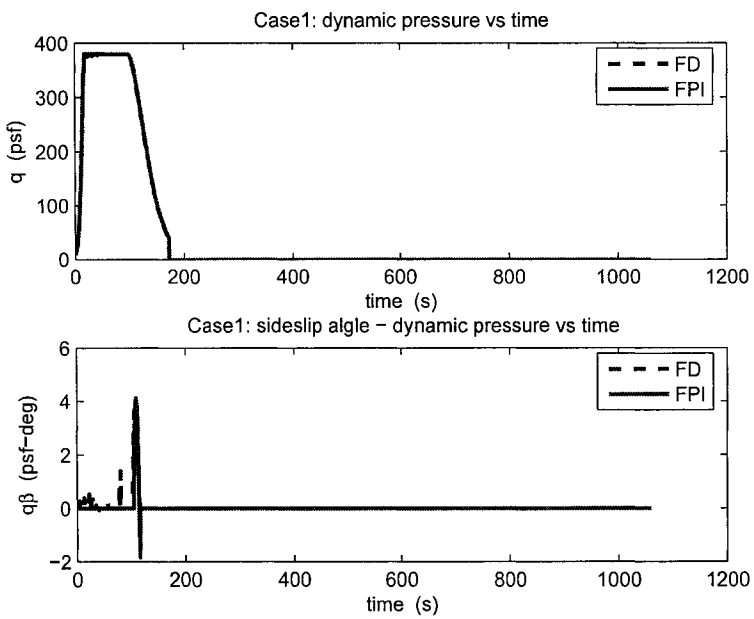

Figure 6.6 Dynamic pressure and sideslip angle - dynamic pressure profiles for Case1

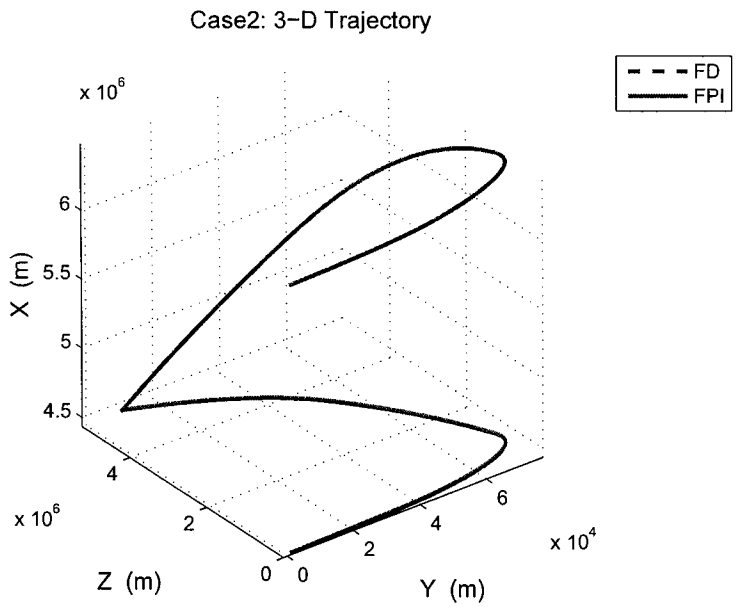

Figure 6.7 Complete 3-D ascent trajectory and ground track for Case2

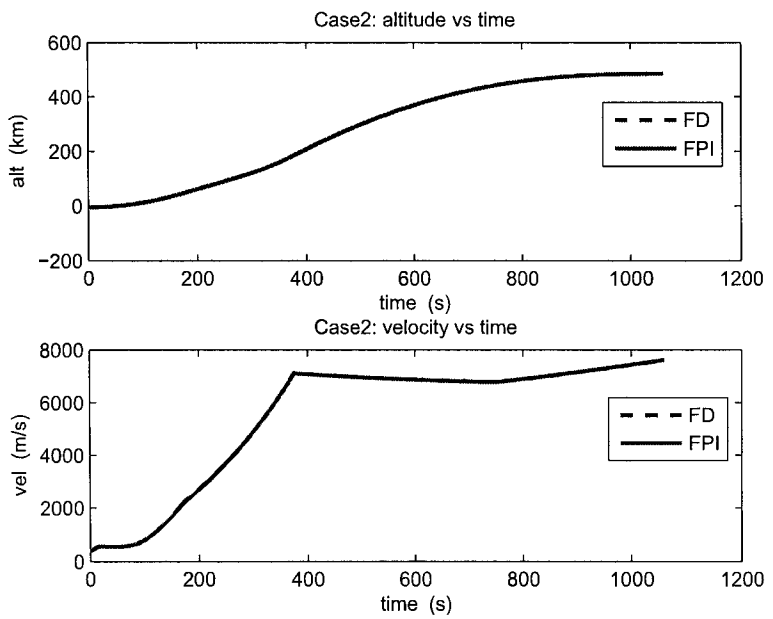

Figure 6.8 Altitude and velocity profiles for Case2

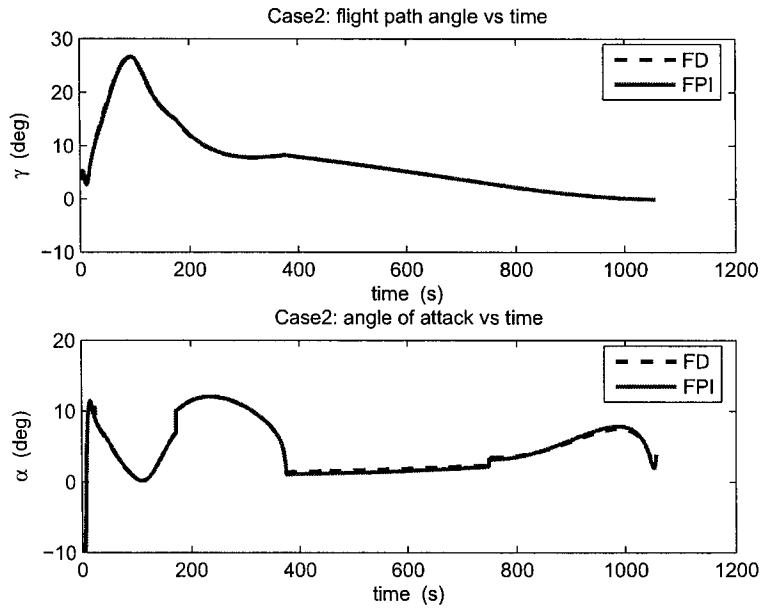

Figure 6.9 Flight path angle and angle of attack profiles for Case2

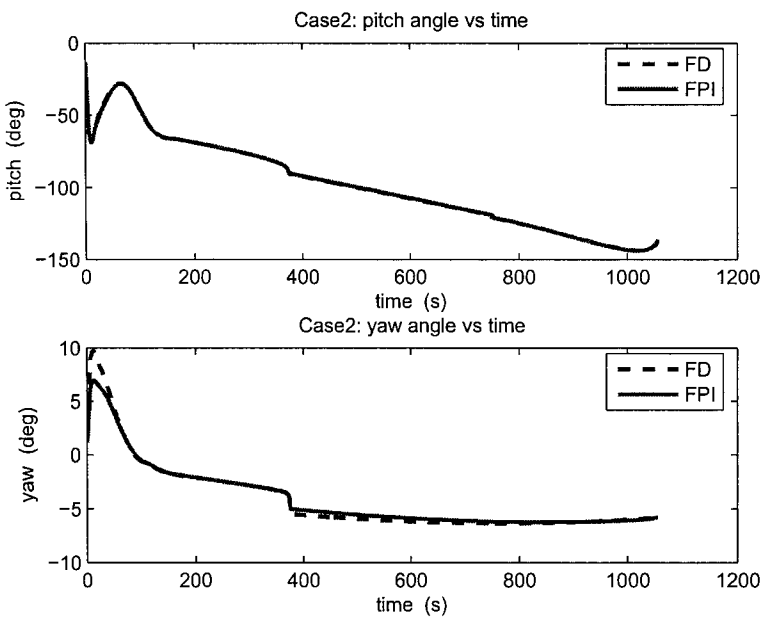

Figure 6.10 Pitch angle and yaw angle profiles for Case2

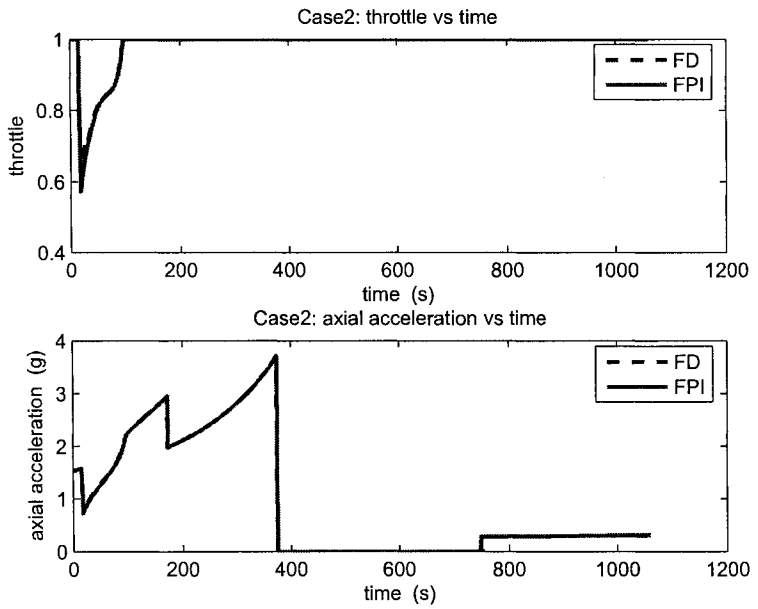

Figure 6.11 Throttle and axial acceleration profiles for Case2

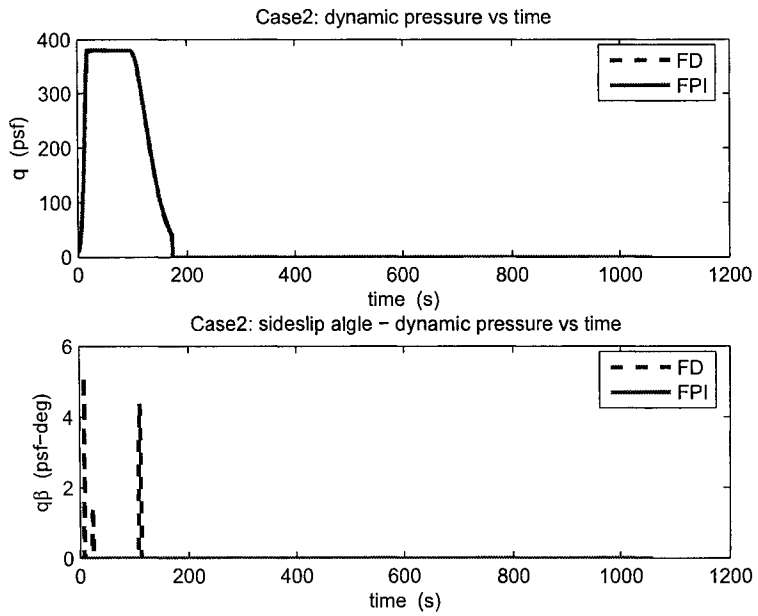

Figure 6.12 Dynamic pressure and sideslip angle - dynamic pressure profiles for Case2

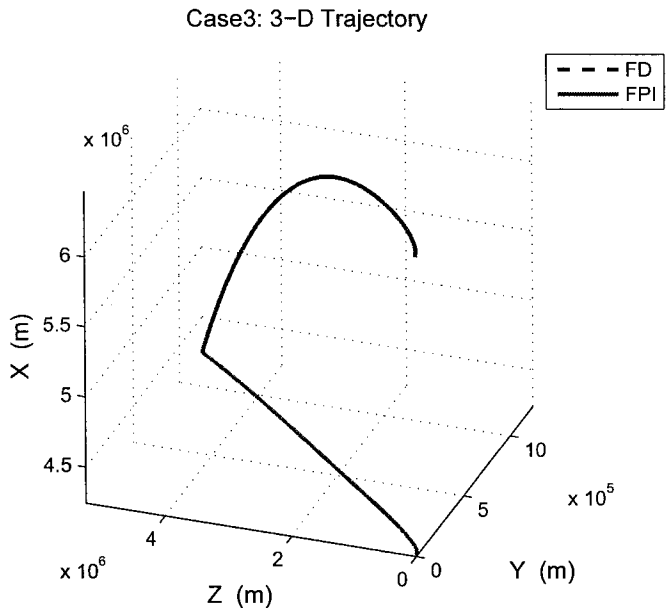

Figure 6.13 Complete 3-D ascent trajectory and ground track for Case3

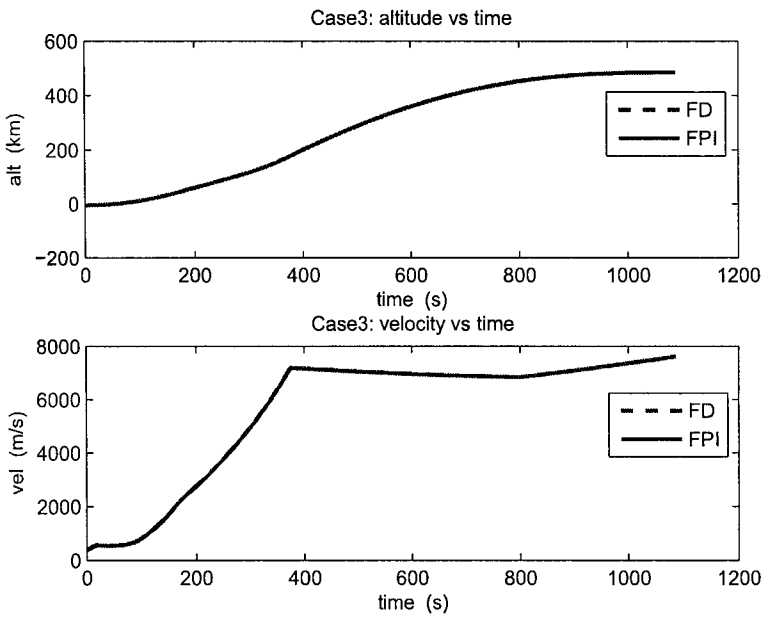

Figure 6.14 Altitude and velocity profiles for Case3

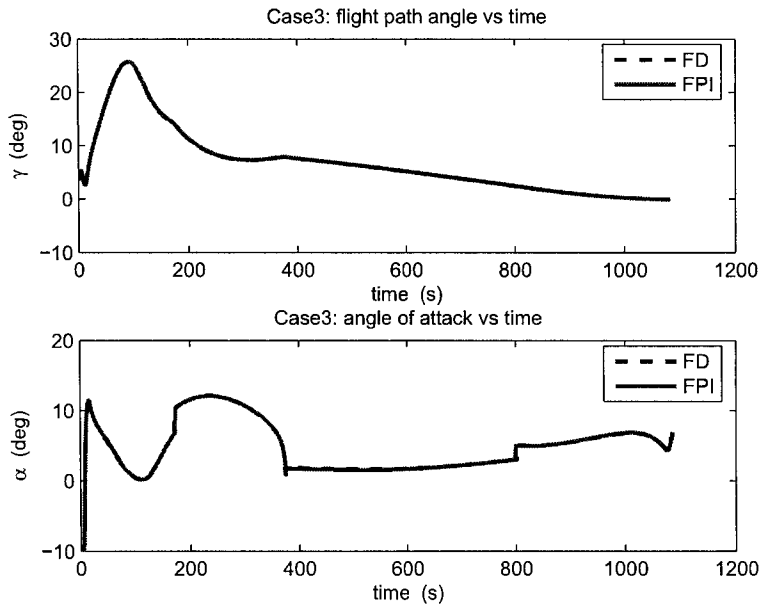

Figure 6.15 Flight path angle and angle of attack profiles for Case3

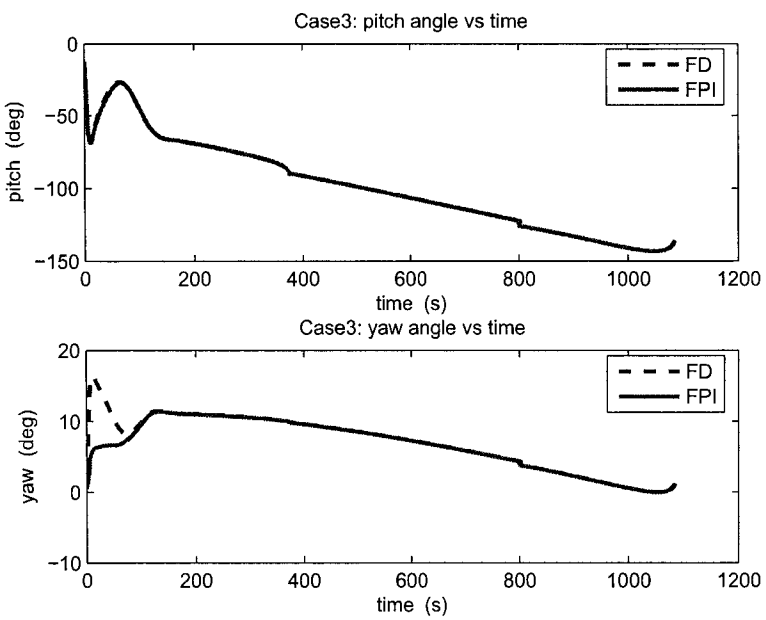

Figure 6.16 Pitch angle and yaw angle profiles for Case3

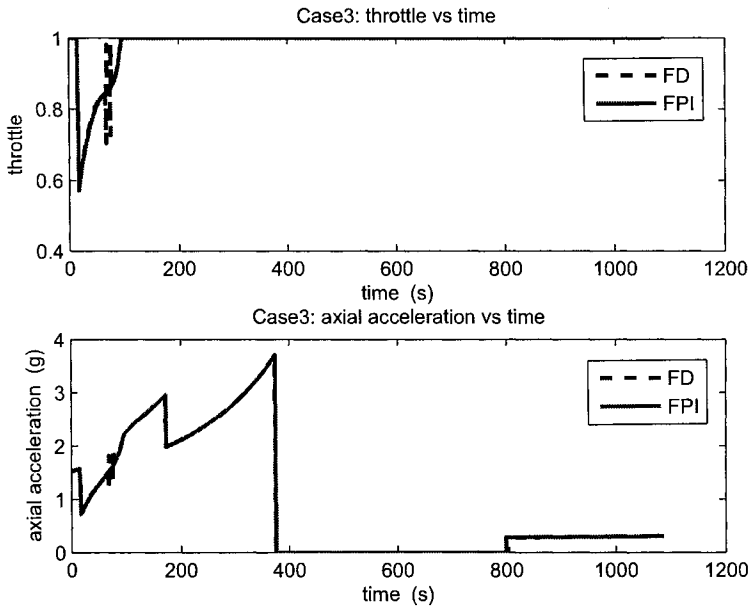

Figure 6.17 Throttle and axial acceleration profiles for Case3

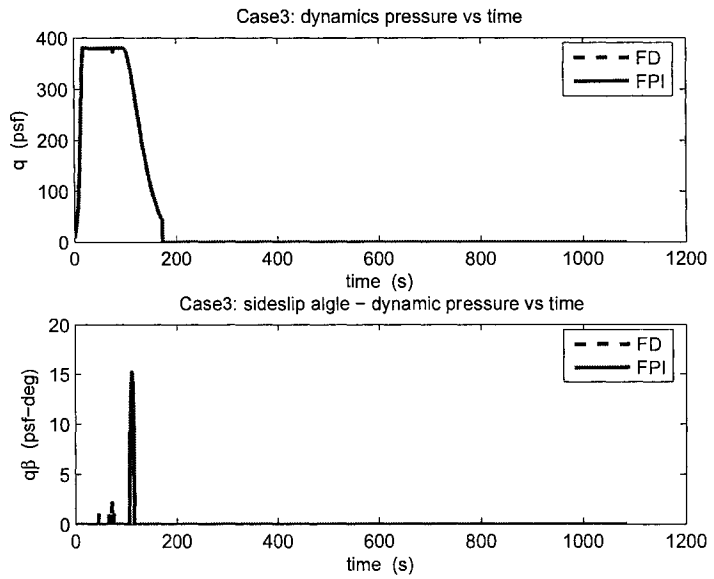

Figure 6.18 Dynamic pressure and sideslip angle - dynamic pressure profiles for Case3

CHAPTER 7. FURTHER COMPARISON BETWEEN FIXED-POINT ITERATION AND FINITE DIFFERENCE ALGORITHMS

From the tables and figures of open-loop simulation and closed-loop simulations for different test cases in Chapter 6, it can be seen that FPI and FD algorithms present almost same results in open-loop and closed-loop simulations. FD algorithm is a well-established numerical algorithm used for solving optimal endo-atmospheric ascent trajectories. Therefore the observation on simulation results confirms the effectiveness of the FPI algorithm developed in this research. In this Chapter, the further comparison between FPI and FD algorithms will be presented.

7.1 Comparison of Programming Simplicity between Fixed-Point Iteration (FPI) and Finite Difference (FD) Algorithms

To solve the special form of TPBVP generated by the algorithmic design, the FPI algorithm consists of one outer-loop iteration on $\{\mathbf{x}_i\}$, and two inner-loop fixed-point iterations (5.11) and (5.12). Therefore the FPI algorithm relies exclusively on fixed-point iterations and the structure of the software implementation is very simple and straight-forward. In the code, the subroutine FPISOLVDE is called to find the nominal optimal endo-atmospheric ascent trajectory. It includes one outer DO WHILE loop

which is used to iterate on $\{\mathbf{x}_i\}$ and two inner nested DO-Loops used respectively for iteration on costate and state variables. Each inner nested DO-Loop includes another DO WHILE loop which is used to execute the fixed-point iteration on costate or state variables at each node. The central difference method is employed to generate the finite difference equation for both of costate and state variables at each node. Therefore only one more subroutine FINITE is required in each inner DO WHILE loop to find the state or costate variables at current node with finite difference equations. Let us suppose the current node index is i, the subroutine FINITE will provide the evaluation of the cosate variable $\mathbf{p}_{i-1}$ or state variable $\mathbf{x}_{i+1}$ for each inner nested DO-Loop. The simplicity of FPI algorithm can be seen from Figure 7.1.

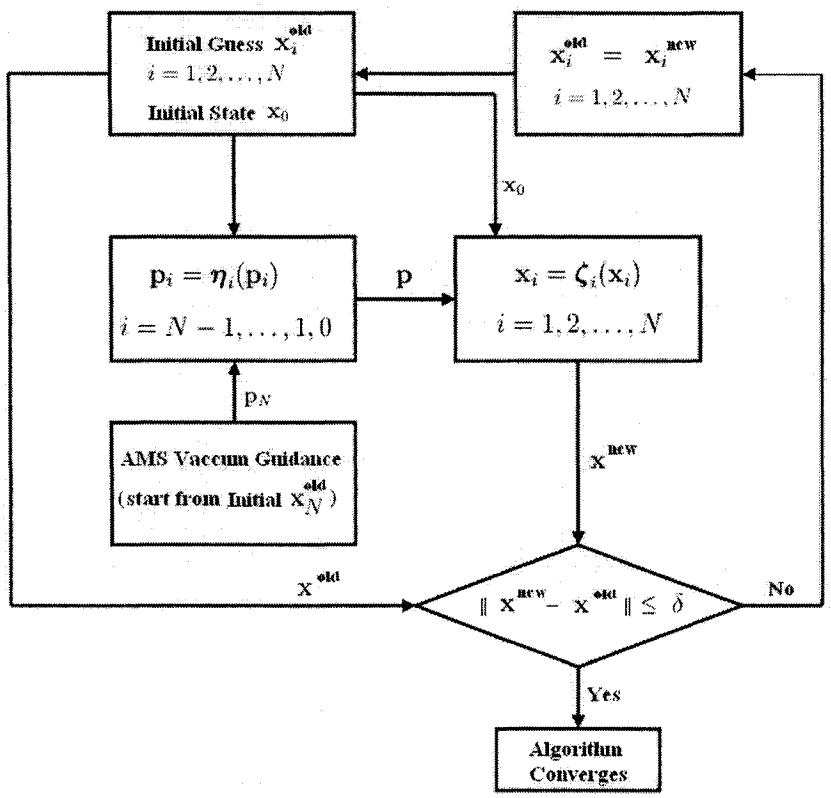

Figure 7.1 Flow chart of FPI algorithm

In comparison to FPI algorithm, the FD algorithm is a little more complicated. FD algorithm solves a root-finding problem for a system of nonlinear algebraic equations constructed by the central finite difference equation at each middle point of two adjacent nodes. It involves the gradient-based steps. The modified Newton method, Gauss elimination and back substitutions are employed to solve the root-finding problem. The code of FD algorithm includes the subroutine FDSOLVD, which is the driver routine for the solution of TPBVP. It will call other five subroutines RED, PINVS, BKSUB, DIFEQ and EVR respectively to solve a root-finding problem. See Ref. [48] for detail.

For the implementation of FPI algorithm, the code is only around 400 lines. For the FD algorithm, the code is around 520 lines.

7.2 Comparison of Efficiency between Fixed-Point Iteration (FPI) and Finite Difference (FD) Algorithms

The algorithm logic of the FPI algorithm is relatively simple. Starting from the initial guess of state (position and inertial velocity vector) at each node along the ascent trajectory, the fixed-point iteration is employed at each node during each outer-loop iteration on state trajectory. Once the outer-loop fixed-point iteration on state trajectory converges, a fixed-point iteration will be performed on the state at the switch point between atmospheric flight and vacuum flight. Several approaches have been developed to facilitate or accelerate the convergence of the inner-loop and outer-loop fixed-point iterations. This section compares computational efficiency of the FPI with that of the FD algorithm.

Let us choose the time needed for executing the open-loop and closed-loop simulation from lift-off to orbital insertion in a specific mission as the criterion of algorithm efficiency. The performed missions are Mission 6, Mission 7 and Mission 8 in Chapter 6.

From Table 7.1, it can be seen that FD algorithm is more efficient in comparison

Table 7.1 Comparison of Algorithm Efficiency between FPI and FD

	method	Mission 6	Mission 7	Mission 8
open-loop (sec.)	FD	4.0	6.0	5.0
	FPI	9.0	15.0	10.0
closed-loop (sec.)	FD	145.0	248.0	126.0
	FPI	246.0	575.0	293.0
time-to-go (sec.)	FD	1037.9	939.7	921.5
	FPI	1033.8	937.9	917.9

to FPI algorithm in terms of the time required to complete the open-loop and closed-loop simulations for a specific mission. This result should not be a surprise because gradient-based algorithms generally converge faster than nongradient algorithms.

7.3 Comparison of Sensitivity to Initial Guesses between Fixed-Point Iteration (FPI) and Finite Difference (FD) Algorithm

The sensitivity with respect to initial guesses is one of the critical points in evaluating the quality of ascent guidance algorithm. For the FPI and FD endo-atmospheric ascent guidance algorithms, the required initial guess is the position and inertial velocity profiles along the ascent trajectory. It is easy to find an optimal ascent trajectory with the vacuum ascent guidance algorithm. Therefore in all open-loop and closed-loop simulations presented in Chapter 6, the initial guesses for different test cases come from the vacuum ascent guidance algorithm. Figure 7.2 gives an example about the initial guess of the 3-D trajectory and the converged 3-D ascent trajectories respectively from FPI and FD algorithms.

Obviously the ascent trajectory given by vacuum ascent guidance algorithm is far

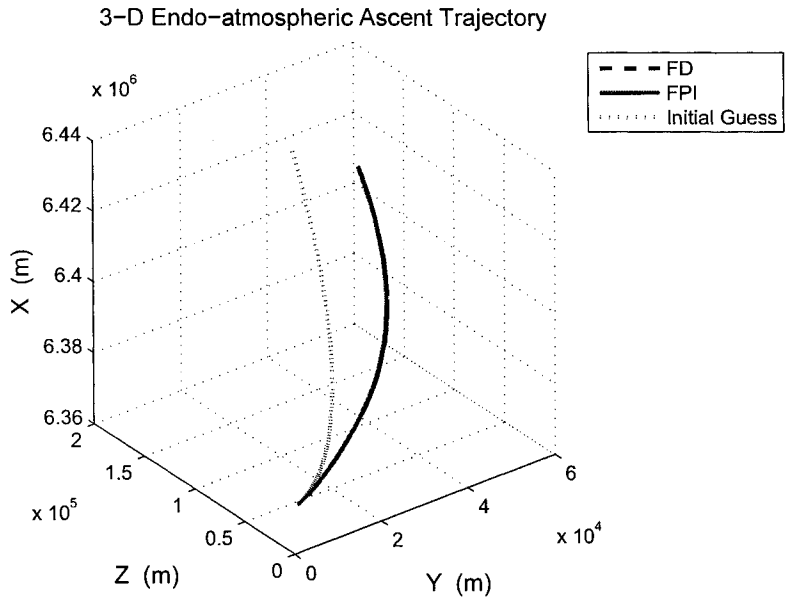

Figure 7.2 A showcase about intial guess of 3-D trajectory and converged 3-D trajectories from FPI and FD

away from the one calculated by the endo-atmospheric ascent guidance algorithm. Such kind of initial guess is called "cold start". It is well known that the poor quality on the initial guess may cause difficulty on convergence of the atmospheric guidance algorithm. Therefore it is always a critical concern for the endo-atmospheric ascent guidance algorithm to have less sensitivity with respect to initial guesses. In the different test cases, it has been found that the FPI algorithm is less sensitive to initial guess than the FD algorithm. In other words, FPI has a better performance than FD algorithm in the "cold start" situations. In following part, simulation examples are given to illustrate this aspect.

Mission 1: circular orbit, orbital inclination i= 51.6 (deg), flight path angle $\gamma = 0$ at insertion point, insertion altitude is 385.2 (km), free ascending node.

Mission 2: elliptical orbit (eccentricity e= 0.02), orbital inclination i= 51.6 (deg), flight path angle $\gamma = 0$ at insertion point, perigee altitude is 347.93 (km), longitude of

ascending node Ω = -104.0 (deg).

Mission 3: elliptical orbit (eccentricity e= 0.01), orbital inclination i= 40. (deg), flight path angle $\gamma = 0$ at insertion point, insertion altitude is 515.57 (km), free ascending node.

All initial guesses are generated by the vacuum ascent guidance algorithm through performing a mission with the defined final altitude r_1, final magnitude of the velocity v_1, final flight path angle γ_1 and inclination i. Since these vacuum solutions are used as the initial guess for the endo-atmospheric ascent portion, the same launch vehicle model is employed. The thrust is 2303000.0 (N) and the mass flow rate is 275.0 (kg/sec). It should be noted that here the final point of the vacuum solution is coincident with the switch point between atmospheric flight and vacuum flight. Since the orbital inclination is 51.6 (deg) for both of *Mission 1* and *Mission 2*, it is highly possible that both of them can share the same initial guess. Therefore the Initial guesses 1 to 8 are generated for the *Mission 1* and *Mission 2*. For the *Mission 3*, Initial guesses 9 to 16 are generated for use. The open-loop simulations are performed for each mission with the different initial guess. It has been found that the Fixed-Point Iteration algorithm can converge successfully when it starts from most of the different initial guesses. However the Finite Difference algorithm encounters difficulty on convergence in many cases. The related information of the test for the *Mission 1*, *Mission 2* and *Mission 3* are summarized respectively in Table 7.2, 7.3 and 7.4.

In Table 7.2, 7.3 and 7.4, the quantities in the column 1 to 4 are the four parameters used for defining the mission performed with the vacuum ascent guidance algorithm. As mentioned before, the solutions from vacuum ascent guidance are used as initial guesses. If the algorithm (FPI and/or FD) can converge successfully when it starts from the initial guess, the name of the algorithm is put into the corresponding 5^{th} column. Elsewise the name of the algorithm is put into the corresponding 6^{th} column.

From Table 7.2, it can be seen that the FPI algorithm can converge from almost each

initial guess except the Initial Guess 6 for performing the *Mission 1*. The FD algorithms works only in one case (Initial Guess 5) but fails in all other cases for performing the *Mission 1*.

From Table 7.3, it can be seen that both of the FPI and FD algorithms converge from most of the different initial guesses. Starting from the Initial Guess 5, FPI algorithm converges successfully but the FD algorithm fails to converge. The flight path angle γ_1, one of the parameters varied in generating the initial guess, is 30 deg in this case. Generally 30 deg is an extreme value of the flight path angle for vehicle's endo-atmospheric flight. So it is conceivable that the quality of Initial Guess 5 is poor. Both of the FPI and FD algorithm fail to converge when they start from the Initial Guess 6 for performing the *Mission 2*. It is probably because of the very low value of the magnitude of v_1 which is one of the parameters varied in generating the initial guess.

The test results for performing the *Mission 3* are listed in Table 7.4. It is found that the FPI algorithm can always get to converge when it starts from eight different initial guesses respectively. However the FD algorithm fails in most of the cases except when it starts from the Initial Guess 9 or Initial Guess 12.

Table 7.2 Test case 1 for comparison of sensitivity with respect to initial guess

Mission 1	r_1 (km)	v_1 (m/sec)	γ_1 (deg)	i	work	fail
Initial Guess 1	45	2300	15	45	FPI	FD
Initial Guess 2	45	2500	5	51	FPI	FD
Initial Guess 3	50	2200	25	51	FPI	FD
Initial Guess 4	50	2400	22	45	FPI	FD
Initial Guess 5	50	2400	30	47	FPI/FD	
Initial Guess 6	50	2000	15	45		FPI/FD
Initial Guess 7	55	2200	20	50	FPI	FD
Initial Guess 8	60	2400	25	45	FPI/FD	

Part of the closed-loop simulation results for performing the *Mission 1* are given in Figure 7.4 to 7.9. All of them only contain the simulation results for endo-atmospheric

Table 7.3 Test case 2 for comparison of sensitivity with respect to initial guess

Mission 2	r_1 (km)	v_1 (m/sec)	γ_1 (deg)	i	work	fail
Initial Guess 1	45	2300	15	45	FPI/FD	
Initial Guess 2	45	2500	5	51	FPI/FD	
Initial Guess 3	50	2200	25	51	FPI/FD	
Initial Guess 4	50	2400	22	45	FPI/FD	
Initial Guess 5	50	2400	30	47	FPI	FD
Initial Guess 6	50	2000	15	45		FPI/FD
Initial Guess 7	55	2200	20	50	FPI/FD	
Initial Guess 8	60	2400	25	45	FPI/FD	

Table 7.4 Test case 3 for comparison of sensitivity with respect to initial guess

Mission 3	r_1 (km)	v_1 (m/sec)	γ_1 (deg)	i	work	fail
Initial Guess 9	45	2200	20	36	FPI/FD	
Initial Guess 10	45	2200	18	40	FPI	FD
Initial Guess 11	50	2300	15	38	FPI	FD
Initial Guess 12	50	2400	20	45	FPI/FD	
Initial Guess 13	55	2200	30	45	FPI	FD
Initial Guess 14	55	2500	10	50	FPI	FD
Initial Guess 15	55	2200	20	40	FPI	FD
Initial Guess 16	60	2300	10	38	FPI	FD

portion. it is easily found that the FPI algorithm presents almost identical closed-loop simulation results when it starts from Initial Guesses 3, 5 and 8. The closed-loop simulation result of the FD algorithm is also provided when it starts from Initial Guess 8. Although the FD algorithm fails to converge in many cases where the FPI algorithm works well, the FPI and FD algorithms yield basically the same closed-loop simulation results in the case where both of them converge successfully.

The evidence presented in this section suggests that the FPI algorithm is considerably more robust with respect to the quality of the initial guesses than FD algorithm is.

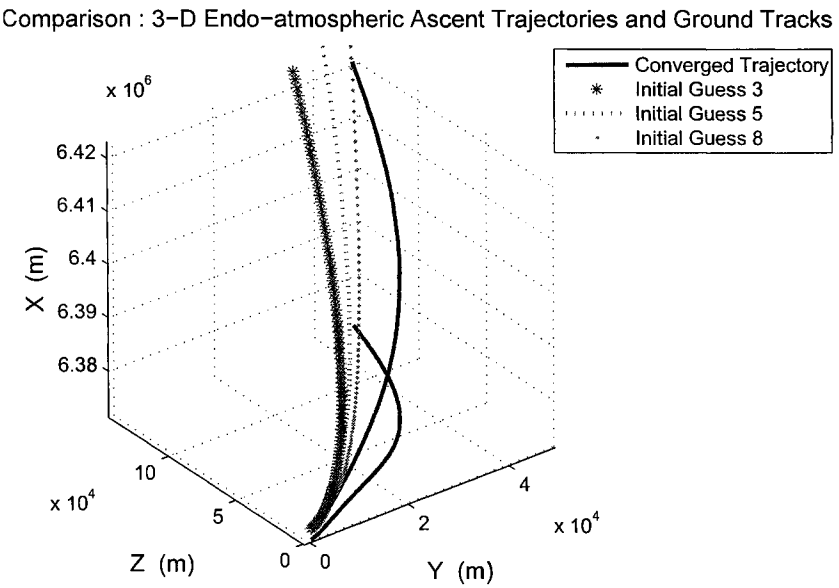

Figure 7.3 3-D endo-atmospheric ascent trajectory and different initial guesses

Starting from different initial guesses (see Figure 7.3), the closed-loop simulation of complete ascent for *Mission 1* is performed. Part of the results are listed in Table 7.5, where τ_1 is the instant switching from atmospheric flight to vacuum flight; $\tau_{meco} - \tau_1$ is the duration of 2^{nd} powered stage; $\tau_f - \tau_{oms}$ is the duration of 3^{rd} powered stage; τ_f is the total time to go. All previous terms are in the unit of *sec.*. $\text{m}(\tau_f)$ is the final mass of the vehicle. It can be seen that the Fixed-Point Iteration algorithm presents almost identical closed-loop simulation results when it starts from different initial guesses.

Table 7.5 Comparison of the closed-loop simulation results with Fixed-Point Iteration algorithm from different initial guesses

	τ_1	$\tau_{meco} - \tau_1$	coast time	$\tau_f - \tau_{oms}$	τ_f	$\text{m}(\tau_f)$ (kg)
FPI, Initial Guess 3	172.33	202.74	341.21	277.03	993.31	38190.85
FPI, Initial Guess 5	172.25	202.74	341.95	276.68	993.62	38195.69
FPI, Initial Guess 8	172.33	202.74	341.29	277.01	993.38	38191.05
FD, Initial Guess 8	171.91	202.74	345.61	275.70	995.96	38209.07

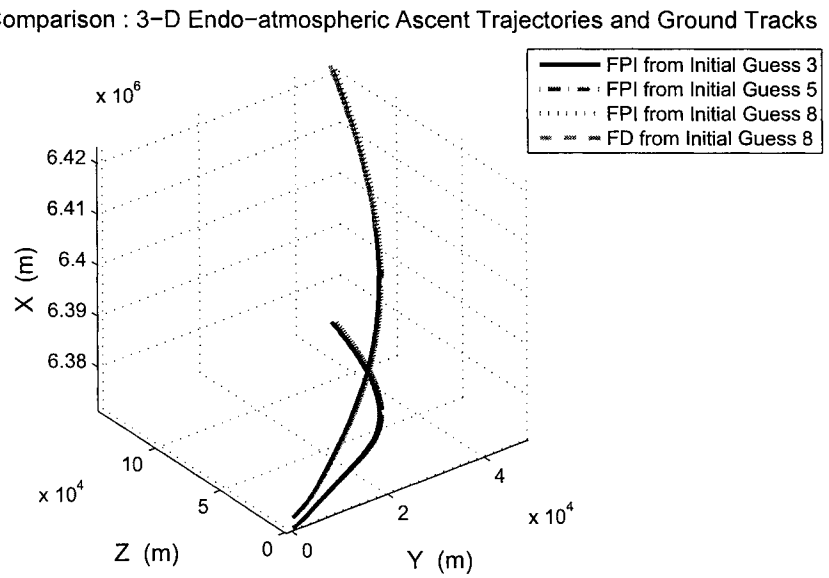

Figure 7.4 3-D endo-atmospheric ascent trajectories and ground tracks

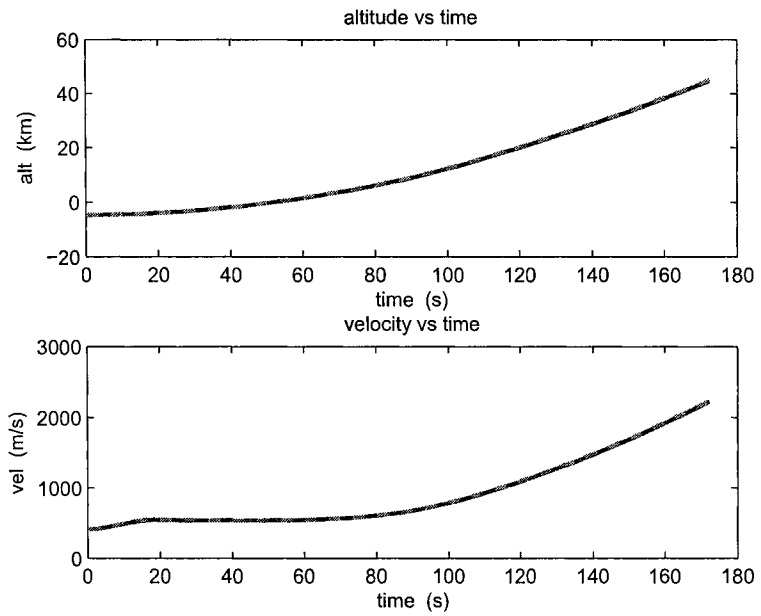

Figure 7.5 Altitude and inertial velocity profiles from different initial guesses in the endo-atmospheric portion

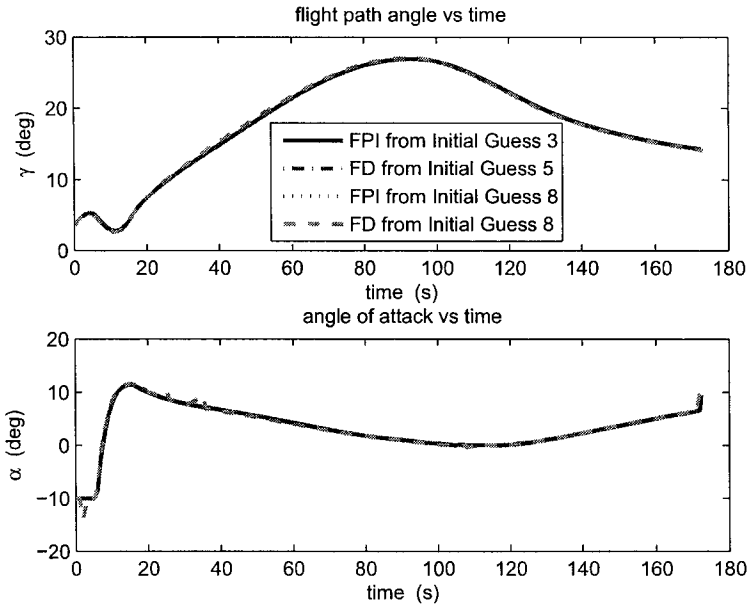

Figure 7.6 Flight path angle and angle of attack profiles from different initial guesses in the endo-atmospheric portion

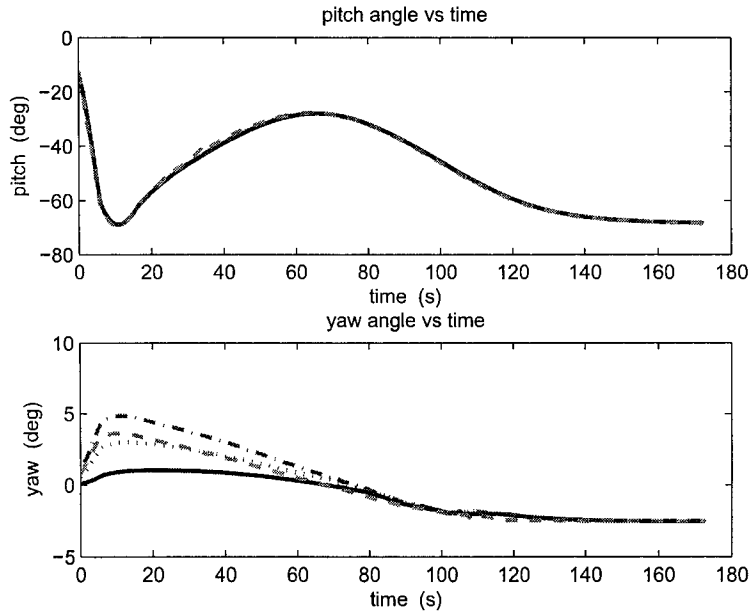

Figure 7.7 Pitch angle and yaw angle profiles from different initial guesses in the endo-atmospheric portion

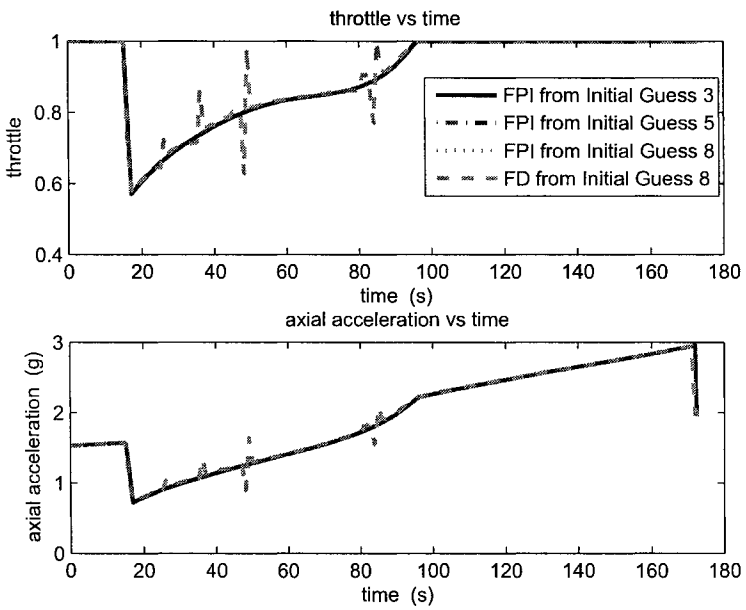

Figure 7.8 Throttle and axial acceleration profiles from different initial guesses in the endo-atmospheric portion

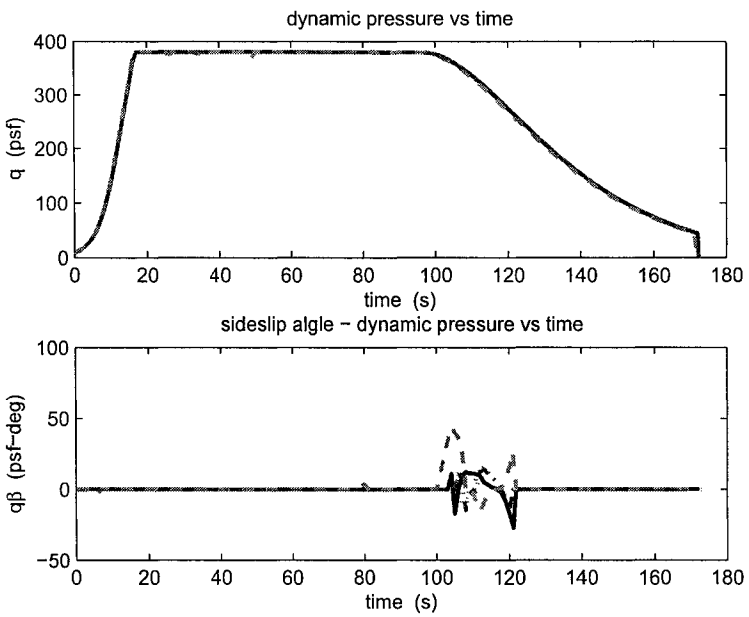

Figure 7.9 Dynamic pressure and sideslip angle-dynamic pressure profiles from different initial guesses in the endo-atmospheric portion

CHAPTER 8. CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

This dissertation takes a unique and unorthodox approach toward developing reliable, fast, and simple ascent guidance algorithms for launch vehicles.

To preserve the simplicity and robustness of vacuum guidance algorithm, we use a fixed-point integration approach to seamlessly combine atmospheric and vacuum ascent guidance algorithms. By exploiting a feature of this integration process and certain special traits of the finite-difference discretization in the atmospheric algorithm, we are able to cast the traditional two-point-boundary-value problem arising from the optimal atmospheric ascent into another nested fixed-point setting. Consequently, the atmospheric ascent guidance algorithm would reduce to a sequence of fixed-point iterations. The corresponding software would be extremely simple. Several scaling techniques are developed to facilitate or accelerate the convergence of the fixed-point iterations. Numerical evidences illustrates clearly the effectiveness of these techniques. The method can be implemented with very simple software, and appears to offer robust convergence. The results suggest promising potential of the presented fixed-point iteration algorithms in rapid launch mission planning, and potentially to closed-loop endo-atmospheric ascent guidance.

Analytical Multiple Shooting algorithm is developed for the burn-coast-burn vacuum ascent trajectory generation and guidance problem. The algorithm keeps the simplicity and robustness of traditional vacuum guidance algorithm through using the classical

analytical solution to the costate equations and approximate analytical solution to the state equations in propagating the state and costate along the vacuum burn arcs. To get a more precise propagation on state variables along coast arc, f and g series method is adopted. The numerical test results verify the effectiveness of presented algorithm in dealing with burn-coast-burn vacuum ascent guidance problem. The developed methodology is readily applicable to the cases with more powered stages and multiple coasts in exo-atmospheric portion.

A series of open-loop and closed-loop simulations for different missions are explored on a generic launch vehicle model. The results demonstrate the effectiveness, robustness and reliability of the methodology, algorithms and numerical techniques developed in this dissertation. Especially the presented Fixed-Point Iteration (FPI) algorithm is more adaptive to the "cold start" case in comparison to the existing Finite Difference (FD) algorithm. Obviously it is a superiority for the autonomy of on-board generation of the optimal ascent trajectories.

8.2 Future Work

Through different tests on open-loop and closed-loop simulations, FPI algorithm shows great robustness and reliability in solving endo-atmospheric ascent optimal trajectories. However the computational efficiency is one of the critical requirements for the on-board generation of optimal ascent trajectories. The simulation results suggest the potential space for further improving the efficiency of this algorithm. On the other hand, the path constraint on the product of dynamic pressure and angle of attack $|q\alpha|$ is not considered in this study. To make the methodology and algorithms developed in this dissertation to be more closed to engineering application in the real world guidance problem of launch vehicles, following further research are recommended.

- Further analysis and numerical techniques are expected to improve the computa-

tion efficiency of FPI algorithm.

In this dissertation, several approaches are developed to facilitate the convergence of the Fixed-Point iterations. Through the tests on closed-loop simulations, it has been seen that the Fixed-Point Iteration algorithm is less efficient in comparison to the Finite Difference algorithm in generation of the optimal endo-atmospheric ascent trajectories. Therefore the more effective techniques are expected to speed up the convergence of the fixed-point iterations.

- The path constraint on the product of dynamic pressure and angle of attack $|q\alpha|$ should be incorporated into the code to further test the robustness and reliability of FPI algorithm.

  Once the path constraint on $|q\alpha|$ is incorporated into the code, it is found that both of FD and FPI algorithm will display heavy sensitivity on initial guess. On the other hand, the path constraint on $|q\alpha|$ is one of the critical points for safety of the launch vehicle during atmospheric flight. Therefore a feasible approach is expected to incorporate the path constraint on $|q\alpha|$ into ascent guidance algorithm without dominant negative effect on robustness, reliability and adaptability of the ascent guidance algorithm.

- Treatment of different types of abort, such as Return To Launch Site (RTLS), Trans-Atlantic Landing (TAL) and Abort To Orbit (ATO) etc., should be added into the algorithm to further improve the adaptability and feasibility of the on-board application.

  The optimal coast is successfully incorporated for vacuum flight portion. It is well known that in some abort cases, such as TAL, it is necessary to incorporate the coast because of limitation on propellant consuming. Therefore the current research provides a solid backup to deal with the more complicated abort scenarios.

APPENDIX A: ANALYTICAL EXPRESSIONS OF THE JACOBIAN

Define

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_v \\ \mathbf{p}_r \end{bmatrix} \tag{8.1}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} \\ \mathbf{V} \end{bmatrix} \tag{8.2}$$

$$\lambda = \begin{bmatrix} \mathbf{p}_v \\ -\frac{\mathbf{p}_r}{\omega} \end{bmatrix} \tag{8.3}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{r} \\ \frac{\mathbf{V}}{\omega} \end{bmatrix} \tag{8.4}$$

where ω is the nondimensional Schuler frequency.

Define the matric Ψ_p and Ψ_x as:

$$\Psi_p = \begin{bmatrix} I_3 & 0 \\ 0 & -\frac{1}{\omega} I_3 \end{bmatrix} \tag{8.5}$$

$$\Psi_x = \begin{bmatrix} I_3 & 0 \\ 0 & \frac{1}{\omega} I_3 \end{bmatrix} \tag{8.6}$$

Therefore, we will have

$$\frac{\partial \boldsymbol{S}_1}{\partial \mathbf{p}_1} = \Psi_p^{-1} \cdot \Phi(\tau_{meco} - \tau_1) \cdot \Psi_p \tag{8.7}$$

$$\frac{\partial \boldsymbol{S}_2}{\partial \mathbf{p}_1} = \Psi_x^{-1} \cdot \Gamma(\tau_{meco} - \tau_1) \cdot \frac{\partial \mathbf{I}_0(\tau_{meco} - \tau_1, 0)}{\partial \lambda_0} \cdot \Psi_p \tag{8.8}$$

where $\frac{\partial \mathbf{I}_0}{\partial \lambda_0}$ will be derived in following part. See the definitions of the Φ and Γ in Section 5.3.1.

Let us define:

$$\lambda(ih) = \left[\begin{array}{cc} I_3 & 0 \end{array} \right]_{3\times 6} \Phi(ih) \tag{8.9}$$

$$= \left[\begin{array}{cc} \cos(\omega ih) \cdot I_3 & \sin(\omega ih) \cdot I_3 \end{array} \right]_{3\times 6} \tag{8.10}$$

$$\mathbf{K}(ih) = \frac{1}{\|\mathbf{p}_V(ih)\|} \left[\begin{array}{cc} I_3 & \frac{\mathbf{p}_V(ih) \cdot \mathbf{p}_V(ih)^T}{\|\mathbf{p}_V(ih)\|^2} \end{array} \right]_{3\times 3} \cdot \lambda(ih) \tag{8.11}$$

where $h = \frac{\tau_{meco} - \tau_1}{4}$, $\mathbf{p}_V(t)$ is propagated from $\mathbf{p}_V(\tau_1)$.

From Eq. (5.71), we have

$$\begin{cases} \dfrac{\partial \mathbf{I}_{c0}(\tau_{meco} - \tau_1, 0)}{\partial \lambda_0} = \dfrac{\tau_{meco} - \tau_1}{90} \displaystyle\sum_{i=0}^{4} b_i T(ih) \cos(\omega ih) K(ih) \\ \dfrac{\partial \mathbf{I}_{s0}(\tau_{meco} - \tau_1, 0)}{\partial \lambda_0} = \dfrac{\tau_{meco} - \tau_1}{90} \displaystyle\sum_{i=0}^{4} b_i T(ih) \sin(\omega ih) K(ih) \end{cases} \tag{8.12}$$

where $b_0 = 7$, $b_1 = 32$, $b_2 = 12$, $b_3 = 32$, $b_4 = 7$. $\frac{\partial \mathbf{I}_0}{\partial \lambda_0}$ is given by:

$$\frac{\partial \mathbf{I}_0(\tau_{meco} - \tau_1, 0)}{\partial \lambda_0} = \left[\begin{array}{c} \frac{\partial \mathbf{I}_{c0}(\tau_{meco} - \tau_1, 0)}{\partial \lambda_0} \\ \frac{\partial \mathbf{I}_{s0}(\tau_{meco} - \tau_1, 0)}{\partial \lambda_0} \end{array} \right] \tag{8.13}$$

$$\frac{\partial \boldsymbol{S}_3}{\partial \mathbf{p}_{meco}} = \Psi_p^{-1} \cdot \Phi(\tau_{oms} - \tau_{meco}) \cdot \Psi_p \tag{8.14}$$

$$\frac{\partial \boldsymbol{S}_3}{\partial \tau_{oms}} = \Psi_p^{-1} \cdot \Omega \cdot \Phi(\tau_{oms} - \tau_{meco}) \cdot \Psi_p \cdot \mathbf{p}_{meco}^{+} \tag{8.15}$$

where

$$\Omega = \begin{bmatrix} 0 & \omega I_3 \\ -\omega I_3 & 0 \end{bmatrix} \tag{8.16}$$

It has been noted in Section 5.3.2 that the state at Main Engine Cut-off (MECO) is obtained via f and g series method.

$$\begin{cases} f = \sum_{n=0}^{\infty} \frac{\tau^n}{n!} F_n \\ g = \sum_{n=0}^{\infty} \frac{\tau^n}{n!} G_n \end{cases} \tag{8.17}$$

Therefore

$$\frac{\partial \boldsymbol{S}_4}{\partial \mathbf{x}_{meco}} = \begin{pmatrix} \frac{\partial \boldsymbol{S}_4}{\partial \mathbf{r}_{meco}} & \frac{\partial \boldsymbol{S}_4}{\partial \mathbf{V}_{meco}} \end{pmatrix} \tag{8.18}$$

where

$$\frac{\partial \boldsymbol{S}_4}{\partial \mathbf{r}_{meco}} = \begin{pmatrix} I_3 \cdot f + \mathbf{r}_{meco} \cdot \frac{\partial f}{\partial \mathbf{r}_{meco}} + \mathbf{V}_{meco} \cdot \frac{\partial g}{\partial \mathbf{r}_{meco}} \\ I_3 \cdot \dot{f} + \mathbf{r}_{meco} \cdot \frac{\partial \dot{f}}{\partial \mathbf{r}_{meco}} + \mathbf{V}_{meco} \cdot \frac{\partial \dot{g}}{\partial \mathbf{r}_{meco}} \end{pmatrix} \tag{8.19}$$

$$\frac{\partial \boldsymbol{S}_4}{\partial \mathbf{V}_{meco}} = \begin{pmatrix} I_3 \cdot g + \mathbf{r}_{meco} \cdot \frac{\partial f}{\partial \mathbf{V}_{meco}} + \mathbf{V}_{meco} \cdot \frac{\partial g}{\partial \mathbf{V}_{meco}} \\ I_3 \cdot \dot{f} + \mathbf{r}_{meco} \cdot \frac{\partial \dot{f}}{\partial \mathbf{V}_{meco}} + \mathbf{V}_{meco} \cdot \frac{\partial \dot{g}}{\partial \mathbf{V}_{meco}} \end{pmatrix} \tag{8.20}$$

$$\frac{\partial \boldsymbol{S}_4}{\partial \tau_{oms}} = \Psi_x^{-1} \cdot \Omega \cdot \Phi(\tau_{oms} - \tau_{meco}) \cdot \Psi_x \cdot \mathbf{x}_{meco}^{+} \tag{8.21}$$

$$\frac{\partial \boldsymbol{S}_5}{\partial \tau_{oms}} = \frac{\partial \boldsymbol{S}_5}{\partial \mathbf{x}_f} \cdot \frac{\partial \mathbf{x}_f}{\partial \tau_{oms}} + \frac{\partial \boldsymbol{S}_5}{\partial \mathbf{p}_f} \cdot \frac{\partial \mathbf{p}_f}{\partial \tau_{oms}} \tag{8.22}$$

where

$$\frac{\partial \mathbf{x}_f}{\partial \tau_{oms}} = \Psi_x{}^{-1} \cdot \frac{\partial \mathbf{y}_f}{\partial \tau_{oms}} \tag{8.23}$$

From Eq. (5.69), $\frac{\partial \mathbf{y}_f}{\partial t_{oms}}$ is given by

$$\frac{\partial \mathbf{y}_f}{\partial \tau_{oms}} = -\Omega \cdot \Phi(\tau_f - \tau_{oms}) \cdot \Psi_x \cdot \mathbf{x}_{oms} + \Gamma(\tau_f) \cdot \frac{\partial \mathbf{I}_1(\tau_f, \tau_{oms})}{\partial \tau_{oms}} \tag{8.24}$$

$$= -\Omega \cdot \Phi(\tau_f - \tau_{oms}) \cdot \Psi_x \cdot \mathbf{x}_{oms} - \Gamma(\tau_f) \cdot \begin{bmatrix} T(\tau_{oms}) \cos(\omega \tau_{oms}) \mathbf{1}_{p_v}(\tau_{oms}) \\ T(\tau_{oms}) \sin(\omega \tau_{oms}) \mathbf{1}_{p_v}(\tau_{oms}) \end{bmatrix} \tag{8.25}$$

$$= -\Omega \cdot \Phi(\tau_f - \tau_{oms}) \cdot \Psi_x \cdot \mathbf{x}_{oms} - \Gamma(\tau_f) \cdot \begin{bmatrix} T(\tau_{oms}) \cos(\omega \tau_{oms}) \mathbf{1}_{p_v}(\tau_{oms}) \\ T(\tau_{oms}) \sin(\omega \tau_{oms}) \mathbf{1}_{p_v}(\tau_{oms}) \end{bmatrix} \tag{8.26}$$

Substitute above equation into Eq. (8.23), therefore

$$\frac{\partial \mathbf{x}_f}{\partial \tau_{oms}} = \begin{bmatrix} -\mathbf{V}_f \\ \omega^2 \mathbf{r}_f - T(\tau_f) \mathbf{1}_{p_v}(\tau_f) \end{bmatrix} \tag{8.27}$$

$\frac{\partial \mathbf{p}_f}{\partial \tau_{oms}}$ in Eq. (8.22) is given by:

$$\frac{\partial \mathbf{p}_f}{\partial \tau_{oms}} = {\Psi_p}^{-1} \cdot \frac{\partial \lambda_f}{\partial t_{oms}} \tag{8.28}$$

where

$$\frac{\partial \lambda_f}{\partial t_{oms}} = -\Omega \cdot \Phi(\tau_f - \tau_{oms}) \cdot \Psi_p \cdot \mathbf{p}_{oms} \tag{8.29}$$

Substitute into Eq. (8.28), finally we have

$$\frac{\partial \mathbf{p}_f}{\partial \tau_{oms}} = \begin{bmatrix} \boldsymbol{P}_{\tau_f} \\ -\omega^2 \mathbf{p}_{V_f} \end{bmatrix} \tag{8.30}$$

$$\frac{\partial \boldsymbol{S}_5}{\partial \tau_f} = \frac{\partial \boldsymbol{S}_5}{\partial \mathbf{x}_f} \cdot \frac{\partial \mathbf{x}_f}{\partial \tau_f} + \frac{\partial \boldsymbol{S}_5}{\partial \mathbf{p}_f} \cdot \frac{\partial \mathbf{p}_f}{\partial \tau_{oms}} \tag{8.31}$$

In similar way, we can find

$$\frac{\partial \mathbf{x}_f}{\partial \tau_f} = \begin{bmatrix} \mathbf{V}_f \\ -\omega^2 \mathbf{r}_f + T(\tau_f) \mathbf{1}_{p_v}(\tau_f) \end{bmatrix} \tag{8.32}$$

and

$$\frac{\partial \mathbf{p}_f}{\partial \tau_f} = \begin{bmatrix} -\boldsymbol{P}_{\tau_f} \\ \omega^2 \mathbf{p}_{V_f} \end{bmatrix} \tag{8.33}$$

$$\frac{\partial \boldsymbol{S}_5}{\partial \mathbf{p}_{oms}} = \frac{\partial \boldsymbol{S}_5}{\partial \mathbf{x}_f} \cdot \frac{\partial \mathbf{x}_f}{\partial \mathbf{p}_{oms}} + \frac{\partial \boldsymbol{S}_5}{\partial \mathbf{p}_f} \cdot \frac{\partial \mathbf{p}_f}{\partial \mathbf{p}_{oms}} \tag{8.34}$$

where

$$\frac{\partial \mathbf{x}_f}{\partial \mathbf{p}_{oms}} = \Psi_x^{-1} \cdot \Gamma(\tau_f - \tau_{oms}) \cdot \frac{\partial \mathbf{I}_1(\tau_f - \tau_{oms}, 0)}{\partial \lambda_f} \cdot \Psi_p \tag{8.35}$$

Similar with Eq. (8.12) and Eq. (8.13), we have

$$\frac{\partial \mathbf{I}_1(\tau_f - \tau_{oms}, 0)}{\partial \lambda_0} = \begin{bmatrix} \frac{\partial \mathbf{I}_{c1}(\tau_f - \tau_{oms}, 0)}{\partial \lambda_f} \\ \frac{\partial \mathbf{I}_{s1}(\tau_f - \tau_{oms}, 0)}{\partial \lambda_f} \end{bmatrix} \tag{8.36}$$

$$\begin{cases} \dfrac{\partial \mathbf{I}_{c1}(\tau_f - \tau_{oms}, 0)}{\partial \lambda_f} = \dfrac{\tau_f - \tau_{oms}}{90} \displaystyle\sum_{i=0}^{4} b_i T(ih) \cos(\omega ih) K(ih) \\ \dfrac{\partial \mathbf{I}_{s1}(\tau_f - \tau_{oms}, 0)}{\partial \lambda_f} = \dfrac{\tau_f - \tau_{oms}}{90} \displaystyle\sum_{i=0}^{4} b_i T(ih) \sin(\omega ih) K(ih) \end{cases} \tag{8.37}$$

$\frac{\partial \mathbf{p}_f}{\partial \mathbf{p}_{oms}}$ is given by:

$$\frac{\partial \mathbf{p}_f}{\partial \mathbf{p}_{oms}} = \Psi_p^{-1} \cdot \Phi(\tau_f - \tau_{oms}) \cdot \Psi_p \tag{8.38}$$

$$\frac{\partial \boldsymbol{S}_5}{\partial \mathbf{x}_{oms}} = \frac{\partial \boldsymbol{S}_5}{\partial \mathbf{x}_f} \cdot \frac{\partial \mathbf{x}_f}{\partial \mathbf{x}_{oms}} \tag{8.39}$$

where

$$\frac{\partial \mathbf{x}_f}{\partial \mathbf{x}_{oms}} = \Psi_x^{-1} \cdot \Phi(\tau_f - \tau_{oms}) \cdot \Psi_x \tag{8.40}$$

$\frac{\partial \boldsymbol{S}_5}{\partial \mathbf{x}_f}$ and $\frac{\partial \boldsymbol{S}_5}{\partial \mathbf{p}_f}$ in Eq.(8.22), Eq.(8.31), Eq.(8.34) and Eq.(8.39) can be derived from explicit terminal conditions.

$$\frac{\partial \boldsymbol{S}_6}{\partial \mathbf{p}_{oms}} = \mathbf{x}_{oms}^T \cdot \Psi_x^T \cdot \Psi_p \tag{8.41}$$

$$\frac{\partial \boldsymbol{S}_6}{\partial \mathbf{x}_{oms}} = \mathbf{p}_{oms}^T \cdot \Psi_p^T \cdot \Psi_x \tag{8.42}$$

APPENDIX B: THE AERODYNAMICS COEFFICIENTS

$$C_{A0} = \frac{N_0 + N_2 M_a^2 + N_4 M_a^4 + N_6 M_a^6 + N_8 M_a^8}{1.0 + D_2 M_a^2 + D_4 M_a^4 + D_6 M_a^6 + D_8 M_a^8} \tag{8.43}$$

where if $M_a > 10$, then set $M_a = 10$.

Table 8.1 Coefficients in the expression of C_{A0}

k	N_k	D_k
0	0.1010592864758859	n/a
2	-0.2186411287691285	-1.809629011543758
4	0.1879023305695819	1.131570470617568
6	-0.0134492220787003	-0.08187670371847496
8	0.0005239366811309632	0.003499549825499098

$$C_{A1} = \frac{N_0 + N_1 M_a + N_2 M_a^2 + N_3 M_a^3}{1.0 + D_1 M_a + D_2 M_a^2 + D_3 M_a^3 + D_4 M_a^4} \tag{8.44}$$

where if $M_a > 10$, then set $M_a = 10$.

$$C_{A2} = \frac{N_0 + N_2 M_a^2 + N_4 M_a^4 + N_6 M_a^6}{1.0 + D_2 M_a^2 + D_4 M_a^4 + D_6 M_a^6 + D_8 M_a^8} \tag{8.45}$$

where if $M_a > 10$, then set $M_a = 10$.

C_{N0}, C_{N1} and C_{N2} can be found by:

$$C_{N0} = \frac{N_0 + N_2 M_a^2 + N_4 M_a^4 + N_6 M_a^6 + N_8 M_a^8}{1.0 + D_2 M_a^2 + D_4 M_a^4 + D_6 M_a^6 + D_8 M_a^8 + D_{10} M_a^{10}} \tag{8.46}$$

where if $M_a > 10$, then set $M_a = 10$.

Table 8.2 Coefficients in the expression of C_{A1}

k	N_k	D_k
0	0.01926187139583902	n/a
1	-0.09687725414389853	-0.8632704546296464
2	0.1471612536592079	-0.9499624788780489
3	-0.08653678894400032	1.140044597003578
4	n/a	-0.02122149396723018

Table 8.3 Coefficients in the expression of C_{A2}

k	N_k	D_k
0	-1.797769381869679	n/a
2	2.663492763832283	0.8381036894894706
4	-1.617484134981833	-4.784623834773783
6	0.4322170386496278	4.475980905367454
8	n/a	-0.03133200766859341

$$C_{N1} = \frac{N_0 + N_1 M_a + N_2 M_a^2 + N_3 M_a^3 + N_4 M_a^4}{1.0 + D_1 M_a + D_2 M_a^2 + D_3 M_a^3 + D_4 M_a^4} \tag{8.47}$$

where if $M_a > 10$, then set $M_a = 10$.

$$C_{N2} = \frac{N_0 + N_2 M_a^2 + N_4 M_a^4 + N_6 M_a^6 + N_8 M_a^8}{1.0 + D_2 M_a^2 + D_4 M_a^4 + D_6 M_a^6 + D_8 M_a^8} \tag{8.48}$$

where if $M_a > 10$, then set $M_a = 10$.

Table 8.4 Coefficients in the expression of C_{N0}

k	N_k	D_k
0	-0.0090158196512648	n/a
2	0.001045713148113318	-2.179931511018316
4	0.02336573885129189	1.207747452517017
6	-0.01508418544783309	-0.02035845638427871
8	-0.0001762259660626334	0.008570275516577088
10	n/a	-4.792601856211596E-05

Table 8.5 Coefficients in the expression of C_{N1}

k	N_k	D_k
0	3.21370924619896	n/a
1	-7.328396710343143	-2.368241505760232
2	5.202293616611917	2.001633239908113
3	-1.311598265049908	-0.9027717479903558
4	0.3032775500891213	0.2899494505209796

Table 8.6 Coefficients in the expression of C_{N2}

k	N_k	D_k
0	-0.1248559946329345	n/a
2	0.1381688043251729	-2.151052418297249
4	-0.1438568443524675	1.300555404439784
6	0.04125285679220543	-0.1359980584150335
8	0.004449787780680164	0.01508798786996793

BIBLIOGRAPHY

[1] The Space Launch Initiative: Technology to Pioneer the Space Frontier, NASA Marshall Space Flight Center, Pub. 8-1250, FS-2001-06-12-MSFC, June 2001.

[2] Freeman, D. C., Talay, T. A., and Austin, R. E., Reusable Launch Vehicle Technology Program, *Acta Astronautica*, Vol. 41, No. 11, 1997, pp. 777-790.

[3] Freeman, D. C., Talay, T. A., and Austin, R. E., Single-Stage-to-Orbit – Meeting the Challenge, *Acta Astronautica*, Vol. 38, No. 4-8, 1996, pp. 323-331.

[4] Hanson, J. M., A Plan for Advanced Guidance and Control Technology for 2nd Generation Reusable Launch Vehicles, *AIAA Guidance, Navigation and Control Conference and Exhibit*, Monterey, California, August 5-8, 2002

[5] Force Application and Launch from CONUS (FALCON), *Broad Agency Announcement 03-35, PHASE 1*, DARPA, July 29, 2003, Arlington, VA

[6] McHenry, R. L., Brand, T. J., Long, A. D., Cockrell, B. F., and Thribodeau, J. R. III, Space Shuttle Ascent Guidance, Navigation, and Control, *Journal of the Astronuatical Science*, Vol. XXV11, No. 1, January-March, 1979, pp. 1-38.

[7] Lu, P., Sun, H., and Tsai, B., Closed-loop Endo-Atmospheric Ascent Guidance, *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003, pp. 283-294.

[8] Lu, P., Zhang, L., Sun, H., Ascent Guidance for Responsive Launch: a Fixed-Point Approach, AIAA Paper 2005-6453, Agu. 2005.

[9] Zhang, L., Lu, P., Rapid Planning and Guidance for Ascent through Atmosphere, Accepted by AIAA Guidance, Navigation and Control Conference, Keystone, Colorado, Aug.21-24, 2006

[10] Smith, I. E., General Formulation of the Iterative Guidance Mode, NASA TM X-53414, March 1966.

[11] Brown, K. R., and Johnson, G. W., Real-Time Optimal Guidance, *IEEE Transactions on Automatic Control*, Vol. AC-12, No. 5, 1967, pp. 501-506.

[12] Brown, K. R., Harrold, E. F., and Johnson, G. W., Rapid Optimization of Multiple-Burn Rocket Trajectories, NASA CR-1430, September 1969.

[13] Brown, K. R., Harrold, E. F., and Johnson, G. W., Some New Results on Space Shuttle Atmospheric Ascent Optimization, AIAA Paper NO. 70-978, 1970.

[14] Hanson, J. M., Shrader, M. W., and Cruzen, A., Ascent Guidance Comparisons, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Scottsdale, AZ, August 1-3, 1994.

[15] Bradt, J. E., Jessick, M. V. and Hardtla, J. W., Optimal Gidance for Fture Space Applications, *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, Aug. 17-19, 1987.

[16] Cramer, E. J., Bradt, J. E., and Hardtla, J. W., Launch Flexibility Using NLP Guidance and Remote Wind Sensing, *AIAA Guidance, Navigation and Control Conference*, Portland, OR, Aug. 20-22, 1990.

[17] Kelly, W. D., Formulation of Aerodynamic Quantities for Minimum Hamiltonian Guidance, *AIAA Guidance, Navigation and Control Conference*, Hilton Head Island, SC, Aug. 10-12, 1992.

[18] Leung, M. S. K., and Calise, A. J., Hybrid Approach to Near-Optimal Launch Vehicle Guidance, *Journal of Guidance, Control, and Dynamics*, Vol. 17, No.5, 1994, pp. 881-888.

[19] Calise, A. J., Melamed, N., and Lee, S., Design and Evaluation of a Three-Dimensional Optimal Ascent Guidance Algorithm, *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, 1998, pp. 867-875.

[20] Calise, A. J., Further Improvements to a Hybrid Method for Launch Vehicle Ascent Trajectory Optimization, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2000-4261, Denver, CO, August 14-17, 2000.

[21] Gath, P. F., and Calise, A. J., Optimization of Launch Vehicle Ascent Trajectories with Path Constraints and Coast Arcs, *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, 2001, pp. 296-304.

[22] Dukeman, G. A., Atmospheric Ascent Guidance for Rocket-Powered Launch Vehicles, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2002-4559, Monterey, CA, August 5-8, 2002.

[23] Dukeman, G. A., Calise, A. J., Enhancements to an Atmospheric Ascent Guidance Algorithm, *AIAA Guidance, Navigation and Control Conference*, AIAA-2003-5638, Austin, TX, August 11-14, 2003.

[24] Vinh, N. X., General Theory of Optimal Trajectory for Rocket Flight in a Resisting Medium, *Journal of Optimization Theory and Applications*, Vol. 11, No. 2, 1973, pp.189-202.

[25] Lawden, D. F., Optimal Trajectories for Space Navigation, Butterworth, London, UK, 1963, pp. 54-68.

[26] Cohen, A. O., and Brown, K.R., Real-Time Optimal Guidance for Orbital Maneuvering, *AIAA Journal*, Vol. 10, No. 5, 1972, pp.680-685.

[27] Jezewski, D.J., Optimal Analytical Multiburn Trajectories *AIAA Journal*, Vol. 11, No. 9, 1973, pp.1266-1272.

[28] Tartabini, P. V., Lepsch, R. A., Korte, J. J., and Wurster, K. E., A Multidisciplinary Performance Analysis of a Lifting-Body Single-Stage-to-Orbit Vehicle, *38th Aerospace Sciences Meeting & Exhibit*, Reno, NV, January 10-13, 2000.

[29] Space Shuttle Mission Phases Notes, http://www.starsailor.com/download/mission_manual/document_outline_shuttle_mission_phases.pdf, StarSailor.com, Russell Space Center, 2005. Date retrived: March 1, 2005.

[30] Bryson, A. E., Ho, Y.-C., Applied Optimal Control, Washington: Hemisphere Publishing Corporation, New York, 1975.

[31] Keller, H. B., Numerical Methods for Two-Point-Boundary-Value Problems, Blaisdell Publishing Company, Waltham, MA, 1968, pp. 91-100.

[32] William H., Numerical recipes in FORTRAN: the art of scientific computing, Cambridge University Press, Cambridge, UK, 1992, pp.589-608.

[33] X-33 VentureStar, http://www.fas.org/spp/guide/usa/launch/x-33.htm, FAS.com, Federation of American Scientists. Date retrieved: March 15, 2005.

[34] X-33 Summary, http://www.spaceandtech.com/spacedata/rlvs/x33_sum.shtml, SpaceAndTech.com, 2001. Date retrieved: March 15, 2005.

[35] Lockheed Martin X-33 Reusable Launch Vehicle Prototype, http://www.aerospaceweb.org/aircraft/research/x33/index.shtml, AircraftMuseum.com, 2001. Date retrieved: March 15, 2005.

[36] Betts, J. T., and Huffman, W. P., Sparse Optimal Control Software: Version5.0, The Boeing Company, Seattle, WA, June 2000.

[37] Ross, I. M., and Fahroo, F., User's Manual for DIDO 2001 (α): A $MATLAB^{TM}$ Application Package for Dynamic Optimization, Technical Report NPS-AA-01-003, October 2001, Naval Postgraduate School, Monterey, CA.

[38] Bucher, K. R. II, Pierson, B. L., A Perturbation Technique Applied to an Optimal Re-entry Control Problem, *Astronautica Acta.*, Vol. 17, 1972, pp. 239-244.

[39] McAdoo, S. F., Jezewski, D. J., and Dawkins, G. S., Development of a Method for Optimal Maneuver Analysis of Complex Space Missions, NASA TN D-7882, April, 1975.

[40] Stoer, J., and Bulirsch, R., Introduction to Numerical Analysis, Springer-Verlag, 1980, pp. 256-262.

[41] Sun, H., Closed-loop endo-atmospheric ascent guidance, Ph.D. Dissertation, Department of Aerospace Engineering at Iowa State University, 2005

[42] Dukeman, G. A., Closed-Loop Nominal and Abort Atmospheric Ascent Guidance for Rocket-Powered Launch Vehicles, Ph.D. Dissertation, Department of Aerospace Engineering at Georgia Institute of Technology, 2005

[43] Robbins, H. M., Optimality of Intermediate-Trust Arcs of Rocket Trajectories, *AIAA Journal*, Vol. 3, No. 6, 1965, pp. 1094-1098.

[44] Naylor, A. W., and Sell, G. R., Linear Operator Theory in Engineering and Science, Springer-Verlag, New York, 1982, 126–127.

[45] Isaacson, E., and Keller, H. B., Analysis of Numerical Methods, Dover Publications, Inc., Mineola, NY, 1994, pp. 120–123, and pp. 433–434.

[46] Hämmerlin G., and Hoffmann, K.-H., Numerical Mathematics, Springer-Verkag New York, Inc., New York, 1991, pp. 330–337.

[47] Chobotov, V. A., Orbital Mechanics, American Institute of Aeronautics and Astronautics, Inc., Reston, Virginia, 1996, pp. 59–60.

[48] Sun, H., User's Manual for AscentGuidance: A $FORTRAN^{TM}$ Application Package for Ascent Trajectory Optimization, Department of Aerospace Engineering at Iowa State University, August, 2005